Frontiers
in
Artificial
Intelligence
and
Applications

FORMAL ONTOLOGY IN INFORMATION SYSTEMS

Proceedings of the Twelfth International Conference (FOIS 2021)

Edited by
Fabian Neuhaus
Boyan Brodaric

IOS Press

FORMAL ONTOLOGY IN INFORMATION SYSTEMS

Formal Ontology in Information Systems (FOIS) is the flagship conference of the International Association for Ontology and its Applications, a non-profit organization promoting interdisciplinary research and international collaboration at the intersection of philosophical ontology, linguistics, logic, cognitive science, and computer science.

This book presents the 11 papers accepted for the 12th edition of FOIS. The conference was held from 13-17 September 2021 in Bozen-Bolzano, Italy, as a hybrid event with some participants attending on-site in Bolzano and others attending virtually online. The papers are divided into 3 sections and cover a wide range of topics: (1) Foundations, addressing fundamental issues; (2) Applications and Methods, presenting novel uses, systems, tools, and approaches; and (3) Domain Ontology, describing well-formed ontologies in particular subject areas.

9 781643 682488

ISBN 978-1-64368-248-8 (print)
ISBN 978-1-64368-249-5 (online)
ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

FORMAL ONTOLOGY IN INFORMATION SYSTEMS

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, selected doctoral dissertations, handbooks and proceedings volumes. The FAIA series contains several sub-series, including 'Information Modelling and Knowledge Bases' and 'Knowledge-Based Intelligent Engineering Systems'. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. The series has become a highly visible platform for the publication and dissemination of original research in this field. Volumes are selected for inclusion by an international editorial board of well-known scholars in the field of AI. All contributions to the volumes in the series have been peer reviewed.
The FAIA series is indexed in ACM Digital Library; DBLP; EI Compendex; Google Scholar; Scopus; Web of Science: Conference Proceedings Citation Index – Science (CPCI-S) and Book Citation Index – Science (BKCI-S); Zentralblatt MATH.

Volume 344

Recently published in this series

Vol. 343. M. Tropmann-Frick, H. Jaakkola, B. Thalheim, Y. Kiyoki and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXXIII
Vol. 342. P. Hitzler and M.K. Sarker (Eds.), Neuro-Symbolic Artificial Intelligence: The State of the Art
Vol. 341. A.J. Tallón-Ballesteros (Ed.), Modern Management based on Big Data II and Machine Learning and Intelligent Systems III – Proceedings of MMBD 2021 and MLIS 2021
Vol. 340. A.J. Tallón-Ballesteros (Ed.), Fuzzy Systems and Data Mining VII – Proceedings of FSDM 2021
Vol. 339. M. Villaret, T. Alsinet, C. Fernández and A. Valls (Eds.), Artificial Intelligence Research and Development – Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence
Vol. 338. C. Frasson, K. Kabassi and A. Voulodimos (Eds.), Novelties in Intelligent Digital Systems – Proceedings of the 1st International Conference (NIDS 2021), Athens, Greece, September 30 – October 1, 2021

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

Formal Ontology in Information Systems

Proceedings of the Twelfth International Conference (FOIS 2021)

Edited by

Fabian Neuhaus

Otto von Guericke University Magdeburg, Germany

and

Boyan Brodaric

Geological Survey of Canada, Canada

IOS Press

Amsterdam • Berlin • Washington, DC

ISBN 978-1-64368-248-8 (print)
ISBN 978-1-64368-249-5 (online)
Library of Congress Control Number: 2021952104
doi: 10.3233/FAIA344

Publisher
IOS Press BV
Nieuwe Hemweg 6B
1013 BG Amsterdam
Netherlands
fax: +31 20 687 0019
e-mail: order@iospress.nl

For book sales in the USA and Canada:
IOS Press, Inc.
6751 Tepper Drive
Clifton, VA 20124
USA
Tel.: +1 703 830 6300
Fax: +1 703 830 2300
sales@iospress.com

PRINTED IN THE NETHERLANDS

Preface

This volume contains papers accepted for the 12th edition of the Formal Ontology in Information Systems conference (FOIS 2021). The conference occurred in hybrid format involving on-site attendance in Bolzano, Italy, as well as virtual attendance online. This hybrid structure was a first for FOIS and proved to be quite successful, with sessions typically involving a mix of on-site and virtual contributions. Another first for FOIS was the integration of content from two conferences, FOIS 2020 and FOIS 2021, due to the COVID-prompted cancellation of the FOIS 2020 live program. As a result, papers accepted for FOIS 2020 were presented at FOIS 2021, with necessary adjustments to both presentation length and format.

As with FOIS 2020, FOIS 2021 occurred in the broader context of a Bolzano Summer of Knowledge event (BoSK 2021). BoSK 2021 included multiple conferences, workshops, and tutorials, all dedicated to knowledge representation. FOIS 2021 itself reflected this breadth, as it consisted of several workshops and tutorials, an Early Career Symposium, as well as dedicated Demonstration and Ontology Showcase seminars, all in addition to the FOIS 2020 and 2021 paper presentations.

Due to experience gained from FOIS 2020, the organizers made several changes to this edition of FOIS, primarily to stimulate a greater variety of contributions. Historically, FOIS has always accepted a broad range of papers. To highlight this diversity, through explicit solicitation, we introduced three different research tracks: a Foundations track for formal and theoretical issues, an Applications and Methods track for novel ontology uses, systems, approaches, and tools, and a Domain Ontology track for original or significant ontologies in a specific domain of interest. As these tracks are quite distinct, authors and reviewers were provided detailed evaluation criteria to clarify expectations. In addition, the Program Committee was nearly doubled to better reflect the diversity of the community. As a minor change, the rebuttal phase of reviewing was retained from FOIS 2020, but somewhat shortened.

Overview of Accepted Papers

As hoped, the expansion of the program committee and the introduction of different research tracks led to increased submissions of papers on applications and methods as well as on domain ontologies. In contrast, compared to FOIS 2020, fewer papers were submitted on foundational topics. There was a continuing trend in all three tracks to social and agent themes.

For FOIS 2021 we accepted 11 of 42 research paper submissions, which is an acceptance rate of 26%. The submissions ranged across a wide variety of topics, as typical for FOIS, distributed across the three research tracks:

- Foundations: 5 accepted
- Applications and Methods: 3 accepted
- Domain Ontology: 3 accepted

Foundations

The foundations track is dominated by papers concerned with representational issues: three papers focus on formalisms for multiple perspectives, concept descriptions, and certain natural language scenarios, and a fourth paper is concerned with the nature of representation itself. In *Standpoint Logic: Multi-Perspective Knowledge Representation*, Gómez Álvarez and Rudolph develop a logic framework for representing multiple perspectives in cases of semantic heterogeneity, with biological examples. The problem of expressing concepts, possibly within evolving or multi-perspective scenarios, is considered by Selway, Stumptner and Mayer in their paper entitled *Towards Formalisation of Concept Descriptions and Constraints.* As a third contribution to this track, Bennett explores logic-based representations for a particular natural language construct in *Semantic Analysis of Winograd Schema No. 1*, concluding representational structures such as ontologies play an important role in advancing machine resolution of the construct. From a more birds-eye view, and in the fourth accepted paper on representational issues, the ontological structure of representation itself is investigated by Mizoguchi and Borgo in *An Ontology of Representation*. The fifth and final paper in this track is by Fumiaki Toyoshima, Adrien Barton, Ludger Jansen and Jean-François Ethier, and is entitled *Towards a Unified Dispositional Framework for Realizable Entities*. It explores the ontological nature of things such as dispositions and roles, analyzed with an eye for potential application within the BFO foundational ontology.

Applications and Methods

Despite a wide variety of papers submitted to this track, the accepted papers fall into two distinct categories: (1) methodological, focused on methods related to ontology-supported concept combination and logical inconsistency, as well as (2) system-oriented, focused on a platform for collaborative ontology design. The contribution by Righetti, Porello, Troquard, Kutz, Hedblom and Galliani, entitled *Asymmetric Hybrids: Dialogues for Computational Concept Combination* presents an ontology-supported and dialogue-based approach to concept combination, in which contributing concepts exert unequal influence on the resulting combination. In *Debugging classical ontologies using defeasible reasoning tools*, Coetzer and Britz explore an approach to finding and resolving logical inconsistencies in ontologies using defeasible reasoning to strategically weaken faulty axioms. The system-oriented paper in this track describes a platform for collaborative ontology design, exemplified by development and refinement of the FIBO ontology for the financial domain in *An infrastructure for collaborative ontology development – Lessons learned from developing the Financial Industry Business Ontology (FIBO)* by Allemang, Garbacz, Grądzki, Kendall and Trypuz.

Domain Ontology

The three accepted papers in the domain ontology track tackle quite different subject matter, albeit all linked somehow to actual or simulated human activity, e.g. healthy living, personal data privacy, and robot action. *NAct: The Nutrition and Active Ontology for healthy living* by Tsatsou, Lalama, Wilson-Barnes, Hart, Cornelissen, Buys, Pagkalos, Dias, Dimitropoulos and Daras, presents an ontology of factors to support healthy living implemented in a decision system. The paper by El Ghosh and Abdulrab entitled *Capturing the Basics of the GDPR in a Well-Founded Legal Domain Modular Ontology* develops and evaluates an ontology to support implementation of European data protection regulations, founded on the UFO ontology. The final paper in this track explores ontological support for robotic action in *Foundations of the Socio-physical Model of Activities (SOMA) for Autonomous Robotic Agents* by Beßler, Porzel, Pomarlan, Vyas, Höffner, Beetz, Malaka and Bateman.

Awards

FOIS 2021 conferred three awards: best paper, distinguished paper, and best student paper. The best paper award was sponsored by IOS Press, and the best student paper award was sponsored by the Artificial Intelligence Journal. The awards were mutually exclusive, as a winner in one category could not win in another, with the best paper award taking precedence. The selection was made difficult, as ever, by a number of high quality candidates.

After much deliberation by the selection committee, the FOIS best paper award was given to Guendalina Righetti, Daniele Porello, Nicolas Troquard, Oliver Kutz, Maria Hedblom and Pietro Galliani for their contribution entitled *Asymmetric Hybrids: Dialogues for Computational Concept Combination*. Submitted to the Applications and Methods track, this paper provides novel insight into the integration of applied ontology and the cognitive science field of conceptual combination.

The distinguished paper award was awarded to Fumiaki Toyoshima, Adrien Barton, Ludger Jansen and Jean-François Ethier for their paper entitled *Towards a Unified Dispositional Framework for Realizable Entities*, which was submitted to the Foundations track. It analyzes realizable entities, such as dispositions and roles, and proposes an enhanced classification.

The best student paper award went to Simone Coetzer and Arina Britz for their entry in the Applications and Methods track entitled *Debugging classical ontologies using defeasible reasoning tools*, which helps identify and rectify logical inconsistencies in ontologies.

Acknowledgements

Authors of all submitted papers, accepted or not, are sincerely thanked for their submissions. These not only enable the conference program to be built, but also serve to keep the conference series robust and current, while bolstering the applied ontology community.

Conferences such as FOIS also rely heavily on the diligent work of the organizing committee, who are especially thanked for their exceptional efforts during the trying circumstances of the COVID pandemic. This includes the general chair (Roberta Ferrario), the chairs of the various events, and the publicity chairs. It also includes members of the program committee, who collectively reviewed all paper submissions in concert with a small number of external reviewers. We would also like to thank Megan Katsumi, the proceedings chair, whose aid was instrumental in the creation of this volume. A full listing of the organizing committee is included after this preface.

A special mention is owed to the local organizers: Oliver Kutz and Nicolas Troquard. The COVID-19 pandemic led to constantly shifting policies by the Free University of Bolzano, as well as by local and national governments. This resulted in restricted and evolving travel conditions and local requirements, making long-term planning nearly impossible. Furthermore, the change to a hybrid live-virtual event led to complex infrastructure situations requiring considerable on-the-fly adjustments. While these factors were a potential recipe for disaster, in the end, and much to the credit of the local organizers, the conference proceeded smoothly and was enjoyed by both on-site and remote participants.

FOIS 2021, like its recent predecessors, was organized under the auspices of the IAOA (International Association for Ontology and its Applications). IAOA not only provides a governance framework for FOIS, but is a source of invaluable guidance during all stages of the conference. We thank IOS Press for its continued support in the publication of the FOIS proceedings and its sponsorship of the best paper award. We also thank the Artificial Intelligence Journal for sponsorship of the best student paper award. The following sponsors are also gratefully acknowledged: the Free University of Bozen-Bolzano as well as its KRDB Research Centre for Knowledge and Data, and the Italian National Lab for Artificial Intelligence and Intelligent Systems.

Final Thoughts

Arguably, organising a conference that involves people from many countries meeting in the Italian Alps in the middle of the COVID-19 pandemic was exceedingly ambitious. The idea was certainly born out of the unjustified hope the pandemic would be over by the autumn of 2021. Our main reason for organising FOIS in 2021, on the heels of FOIS 2020, was the conviction that FOIS live events play a crucial role in the Applied Ontology community. Unfortunately, circumstances dictated the majority of our community was only able to participate remotely. Nonetheless, many of the 35 participants who did meet in Bolzano expressed the same sentiment: after 18 months without travel, and after 18 months in which scientific discourse was mostly exiled to virtual spaces, FOIS 2021 was not just a scientific event, it was also welcomed as a place to meet, debate, and reconnect with colleagues and friends.

One final observation: of the eleven research papers accepted at FOIS 2021, six were written by first authors who are either PhD students or early-career researchers. These include the best paper and the distinguished paper of FOIS 2021. That we have so many talented budding researchers in our community gives hope and optimism for the future of FOIS and Applied Ontology.

Fabian Neuhaus
Boyan Brodaric

FOIS 2021 Conference Organization

General Chair

Roberta Ferrario	ISTC-CNR Trento, Italy

Program Chairs

Boyan Brodaric	Geological Survey of Canada, Canada
Fabian Neuhaus	Otto von Guericke University Magdeburg, Germany

Local Chairs

Oliver Kutz	Free University of Bozen-Bolzano, Italy
Nicolas Troquard	Free University of Bozen-Bolzano, Italy

Publicity Chairs

Maria M. Hedblom	Jönköping University, Sweden
Guendalina Righetti	Free University of Bozen-Bolzano, Italy

Proceedings Chair

Megan Katsumi	University of Toronto, Canada

Workshop and Tutorial Chairs

Torsten Hahmann	University of Maine, USA
Claudio Masolo	ISTC-CNR Trento, Italy

Workshop Proceedings Chair

Emilio Sanfilippo	ISTC-CNR Trento, Italy

Early Career Symposium Chairs

Pawel Garbacz	Catholic University of Lublin, Poland
Selja Seppälä	University College Cork, Ireland

Ontology Showcase Chair

Michael Grüninger	University of Toronto, Canada

Demonstrations Chairs

Robert Hoehndorf	King Abdullah Univ. of Science and Technology, Saudi Arabia
Amanda Vizedom	Credit Suisse, USA

Programme Committee

Mara Abel	Universidade Federal do Rio Grande do Sul, Brazil
Panos Alexopoulos	Textkernel B.V., Netherlands
Joao Paulo Almeida	Federal University of Espirito Santo, Brazil
Mauricio Almeida	Federal University of Minas Gerais, Brazil
Bill Anderson	Instacart, USA
Kenneth Baclawski	Northeastern University, USA
Adrien Barton	Institut de Recherche en Informatique de Toulouse, France
Colin Batchelor	Royal Society of Chemistry, UK
John Bateman	University of Bremen, Germany
Alan Belasco	Securboration, USA
Mike Bennett	Hypercube, UK
Olivier Bodenreider	US National Library of Medicine, USA
Stefano Borgo	ISTC-CNR Trento, Italy
Alessander Botti Benevides	Secretariat of Technology and Innovation Vila Velha, Brazil
Mathias Brochhausen	University of Arkansas for Medical Sciences, USA
Boyan Brodaric	Geological Survey of Canada, Canada
Massimiliano Carrara	University of Padua, Italy
Vinay Chaudhri	Stanford University, USA
Philipp Cimiano	Bielefeld University, Germany
Oscar Corcho	Universidad Politécnica de Madrid, Spain
Mathieu D'Aquin	National University of Ireland Galway, Ireland
Sergio de Cesare	University of Westminster, UK
Richard Dapoigny	LISTIC / Polytech'Savoie, France
Maureen Donnelly	SUNY Buffalo, USA
Christiane Fellbaum	Princeton University, USA
Roberta Ferrario	ISTC-CNR Trento, Italy
Frederik Gailly	Ghent University, Belgium
Antony Galton	University of Exeter, UK
Aldo Gangemi	Università di Bologna & ISTC-CNR, Italy
Pawel Garbacz	Catholic University of Lublin, Poland
Chiara Ghidini	Fondazione Bruno Kessler, Italy
Michael Grüninger	University of Toronto, Canada
Nicola Guarino	ISTC-CNR Trento, Italy
Giancarlo Guizzardi	Free University of Bozen-Bolzano, Italy
Torsten Hahmann	University of Maine, USA
Terry Halpin	INTI International University, Australia
Janna Hastings	Otto von Guericke University Magdeburg, Germany
Maria M. Hedblom	Jönköping University, Sweden
Heinrich Herre	University of Leipzig, Germany
Amanda Hicks	University of Florida, USA
Beth Huffer	Lingua Logica, USA
Haythem Ismail	Cairo University and German University in Cairo, Egypt
Krzysztof Janowicz	University of California, Santa Barbara, USA
Ludger Jansen	University of Rostock, Germany
Clement Jonquet	University of Montpellier, France
Gilles Kassel	University of Picardie Jules Verne, France
C. Maria Keet	University of Cape Town, South Africa
Elisa Kendall	Thematix Partners LLC, USA
Dimitris Kiritsis	EPFL, Switzerland
Oliver Kutz	Free University of Bozen-Bolzano, Italy
Christoph Lange	Fraunhofer Institute for Applied Information Technology FIT, Germany
Frank Loebe	University of Leipzig, Germany
Jim Logan	Dassault Systèmes, USA

Claudio Masolo	ISTC-CNR Trento, Italy
Riichiro Mizoguchi	Japan Advanced Institute of Science and Technology, Japan
Friederike Moltmann	IHPST CNRS, France
Joao Moreira	Univesitait Twente, Netherlands
Till Mossakowski	Otto von Guericke University Magdeburg, Germany
Enrico Motta	The Open University, UK
John Mylopoulos	University of Toronto, Canada
Fabian Neuhaus	Otto von Guericke University Magdeburg, Germany
Leo Obrst	MITRE, USA
Alessandro Oltramari	Bosch Research and Technology Center, USA
Jeffrey Parsons	Memorial University of Newfoundland, Canada
Chris Partridge	University of Westminster & BORO Solutions, UK
Oscar Pastor Lopez	Universitat Politènica de València, Spain
Adam Pease	Infosys, USA
Max Petrenko	Amazon, USA
Daniele Porello	Università di Genova, Italy
María Poveda-Villalón	Universidad Politécnica de Madrid, Spain
Quentin Reul	Vrije Univesiteit Brussel, Belgium
Catherine Roussey	French National Institute for Agriculture, Food, and Environment (INRAE), France
Emilio Sanfilippo	ISTC-CNR Trento, Italy
Simon Schneider	Utrecht University, Netherlands
Stefan Schlobach	Vrije Universiteit Amsterdam, Netherlands
Marco Schorlemmer	Artificial Intelligence Research Institute, IIIA, CSIC, Spain
Lynn Schriml	University of Maryland, USA
Cogan Shimizu	Kansas State University, USA
Gaurav Sinha	Ohio University, USA
Barry Smith	SUNY Buffalo, USA
Dagobert Soergel	SUNY Buffalo, USA
Gem Stapleton	University of Cambridge, UK
Chris Stoeckert	University of Pennsylvania, USA
Veda Storey	Gorgia State University, USA
Markus Stumptner	University of South Australia, Australia
Avril Styrman	The Finnish Society for Natural Philosophy, Finland
Mari Carmen Suárez-Figueroa	Universidad Politécnica de Madrid, Spain
Nicolas Troquard	Free University of Bozen-Bolzano, Italy
Guido Vetere	Università Guglielmo Marconi, Italy
Laure Vieu	IRIT CNRS, France
Amanda Vizedom	Credit Suisse, USA
Ramona Walls	CyVerse, USA
Hans Weigand	Tilburg University, Netherlands
Matthew West	Information Logic & Leeds University, UK
Andrea Westerinen	Nine Points Solutions LLC, USA
Jen Williams	USA
Frank Wolter	University of Liverpool, UK
Bob Young	Loughborough University, UK
Antoine Zimmermann	École des Mines de Saint-Étienne, France

Additional Reviewers

Caua Antunes	Universidade Federal do Rio Grande do Sul Brazil
Fabrício Henrique Rodrigues	Universidade Federal do Rio Grande do Sul, Brazil
Karamjit Kaur	University of South Australia, Australia

This page intentionally left blank

Contents

I. Foundations

II. Applications and Methods

III. Domain Ontology

I. Foundations

This page intentionally left blank

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210367

Standpoint Logic: Multi-Perspective Knowledge Representation

Lucía GÓMEZ ÁLVAREZ and Sebastian RUDOLPH
Computational Logic Group, TU Dresden, Germany

Abstract. Ontologies and knowledge bases encode, to a certain extent, the standpoints or perspectives of their creators. As differences and conflicts between standpoints should be expected in multi-agent scenarios, this will pose challenges for shared creation and usage of knowledge sources.

Our work pursues the idea that, in some cases, a framework that can handle diverse and possibly conflicting standpoints is more useful and versatile than forcing their unification, and avoids common compromises required for their merge. Moreover, in analogy to the notion of *family resemblance concepts*, we propose that a collection of standpoints can provide a simpler yet more faithful and nuanced representation of some domains.

To this end, we present *standpoint logic*, a multi-modal framework that is suitable for expressing information with *semantically heterogeneous* vocabularies, where a *standpoint* is a partial and acceptable interpretation of the domain. Standpoints can be organised hierarchically and combined, and complex correspondences can be established between them. We provide a formal syntax and semantics, outline the complexity for the propositional case, and explore the representational capacities of the framework in relation to standard techniques in ontology integration, with some examples in the *Bio-Ontology* domain.

Keywords. standpoint, perspectives, modal logic, ontology integration

1. Introduction

Natural language terms do not have precise, universally agreed definitions that fix their meanings [1]. Instead, their applicability is unclear in some instances, and it may vary depending on the context and pragmatics of use. In borderline cases, the speaker must make a *semantic commitment*: she must decide whether a term is applicable or not.

When an agent formalises a domain, the resulting conceptual model will also be shaped by this kind of semantic commitments, which are in turn influenced by her own worldview and by pragmatic factors, such as the intended granularity and scope.

Even simple domains like colours lend themselves to this issue: while different speakers may agree on most clear instances of *red*, they may disagree on the existence of certain colours (e.g. *vermilion*) and on systems of classification. For instance, consider two (partial) formalisations coming from different perspectives: an ontology of *colour theory* (*CT*), a discipline within the fine arts tradition, and a 'common-sense' representation of ink colours, used by an online *house painting* business (*HP*).

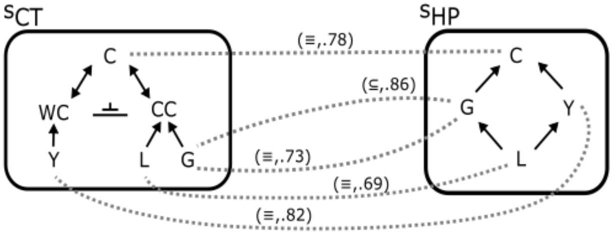


Figure 1. The conceptual models of s_{CT} and s_{HP}, each in a box. Each concept is represented with its initial. In dotted lines one can see the alignments, with their type and similarity in parentheses.

Example 1. It is universally accepted (thus also by CT and HP) that Yellow, Green, and Lime are Colour. According to CT, Colour is either a warm colour (WC) or a cold colour (CC) but not both[1], Yellow is a WC, and Green and Lime are CC. According to HP, Lime is Green and Yellow.

While ontologies are aimed at providing a common vocabulary for representing shared knowledge [2], the described semantic heterogeneity may hinder the interoperation of independently developed systems. Ontology integration is well-studied yet still challenging [3,4,5], and ontology merges often involve certain knowledge loss or weakening in order to avoid incoherence and inconsistency [6,7].

This is the case in Example 1, for which Figure 1 shows the two conceptual models in distinct boxes, as well as the alignments (in dotted lines) that could be found between them using some standard matching algorithm. While simple and intuitively closely aligned, the two conceptualisations cannot be trivially merged without the undesired consequence of Lime becoming unsatisfiable: to prevent this, we must either (1) give up on the JEPD relations of CT, or (2) let go one of the subsumptions of Lime in HP or (3) relinquish one of the alignments, such as Colour, and instead duplicate the concept: Colour_CT and Colour_HP. Yet, it is preferable for the integration to preserve all entailments of the source ontologies and their alignments [8]. Moreover, in areas of growing interest, such as the research on complex alignments and *holistic* (many-source) ontology integration [9,10,11], even more inconsistencies have to be expected.

In this paper, we advocate a multi-perspective approach that can represent and reason with many – possibly conflicting – standpoints, instead of focusing on combining and merging different sources into a single conceptual model.

Beyond the challenges in knowledge integration, we believe that this is also useful in the process of formalising a domain [12]. Typically, heavy axiomatisations enable the derivation of interesting facts, but limit the interoperability of the system; the converse happens with shallow modelling that relies on little more than taxonomic relationships. With our *multi-standpoint* framework, we aim to preserve, on the one hand, the advantages of providing a common high-level conceptual structure for a domain, while, on the other hand, making explicit the more fine-grained semantic commitments associated with different users or interpretations of the domain.

We propose *standpoint logic* (Section 3): a simple formalism rooted in modal logic that supports the coexistence of multiple *standpoints* and the establishment of alignments between them. We highlight that our proposal retains good computational properties: in its

---

[1] These are jointly exhaustive and pairwise disjoint (JEPD) categories of colour.

propositional version, reasoning in the logic is NP-complete (Section 4) in pleasant contrast to the PSPACE-completeness normally exhibited by multi-modal epistemic logics. Then, we proceed to show how the framework allows for the establishment of structures of standpoints and the expression of 'complex alignments' (Section 5), and we illustrate its use in an application in the bio-ontology domain (Section 6). We discuss the background and related work (Section 7) before concluding and providing an outlook on future research (Section 8).

2. Background and Framework Overview

Standpoint Logic is a lightweight multi-modal framework where labelled modalities $\Box_s$ express information relative to a standpoint, and the set of formulae under a standpoint (sentences $\Box_s \phi$ or $\Diamond_s \phi$) represent the agent's world-view or semantic commitments.[2]
Standpoint logic draws from the philosophical theory of supervaluationism, according to which the phenomenon of vagueness (and more generally the semantic variability) can be explained by the fact that natural language can be interpreted in many different yet equally acceptable ways [13], commonly referred to as *precisifications*. Early proposals of this intuition were made by Mehlberg [14], and Fine [15] applied this model to the analysis of vagueness. Supervaluationism is a popular theory of vagueness, adopted by philosophers, logicians and linguists, yet scarce in KR. An exception is the earlier work of Bennett [16], from whom we borrow the notion of standpoint and together we propose a different treatment, already given in [17].
When using a modal infrastructure in a supervaluationistic framework, one replaces the usual structure of *possible worlds* by one of *precisifications*. To see the difference, consider a situation modelled with doxastic logic: Bob believes that there is a red apple at home ($B_{Bob}[\mathsf{red\_apple}]$) and Tim believes that there is a yellow apple instead, ($B_{Tim}[\mathsf{yellow\_apple}]$). Here, the possible worlds model the different possible states of affairs, and there is an *actual world* that dictates what is contingently true. In contrast, consider Tim and Bob facing a red/yellowish apple. Tim calls it a red apple ($\Box_{Tim}[\mathsf{red\_apple}]$), yet Bob calls it a yellow apple, ($\Box_{Bob}[\mathsf{yellow\_apple}]$). The latter is what standpoint logic models, where agents describe the state of the matter using different and equally acceptable interpretations of the vocabulary, so there is no 'actual precisification'.
Different modal frameworks have been proposed in the supervaluationist literature [18, 19,20]. These focus on proposing modalities that can capture the linguistic behaviour of philosophical vagueness and the sorites paradox, and on the analysis of different forms of validity and logical consequence. In contrast, we use the supervaluationist model of natural language (in terms of a collection of admissible classical interpretations) but we focus on scenarios where different agents or different contexts are linked to different usages of such a semantically variable language.
Standpoints are modelled as non-empty sets of *precisifications*, which corresponds to the intuition that a standpoint is (typically) a *partial* semantic commitment that can be made fully precise in different ways. Practical uses of *standpoint logic* include, for instance, the representation of multiple (and possibly conflicting) symbolic conceptualisations of

---

[2] For clarity and better readability, we will sometimes use square brackets $[\cdots]$ to explicitly indicate the scope of the modal operators.

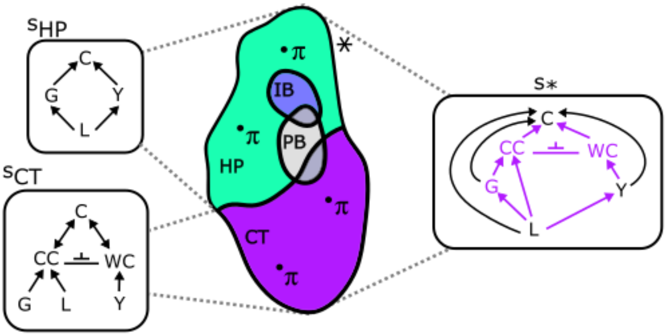


Figure 2. In the center, the precisification space, labelled with $*$, and the standpoints s_{CT}, s_{HP}, s_{PB} and s_{IB} (subsets of $*$). On the sides, the conceptual models of s_{CT}, s_{HP}, and $*$.

a domain, knowing what can be inferred from the consensual semantics of sets of agents and knowing which standpoints are compliant with a partial truth.

In Fig. 2 we can see a representation of the Example 1 in a standpoint framework. The shape in the centre, labelled with $*$, represents the set of all admissible precisifications, where the points (labelled π) are individual precisifications and the subsets (subareas of $*$) are the standpoints s_{CT}, s_{HP}, s_{PB} and s_{IB}. In boxes, on the sides, one can see the corresponding conceptual models that hold for s_{CT}, s_{HP} and s_*, linked with dotted lines to the relevant subset of precisifications that satisfies them. For instance, from the box s_{HP} we can see that $\Box_{HP}[\mathsf{L} \rightarrow (\mathsf{G} \wedge \mathsf{Y})]$, which means that $\mathsf{L} \rightarrow (\mathsf{G} \wedge \mathsf{Y})$ will hold in all the precisifications that belong to the area HP. In the box of s_*, diamond statements are represented in a lighter colour, and generally arrows stand for implications.

In terms of syntax, the standpoint framework exhibits a reasonable versatility through a very economic extension of the (in this case: propositional) base language by constants s for denoting standpoints – with the special constant $*$ denoting the global standpoint – and corresponding pairs of dual modal operators $\Box_s$ and $\Diamond_s$.

$\Box_s \phi$ (or $\Box_* \phi$) reads as “It is unequivocal, [from the point of view s,] that ϕ”. This is the paradigmatic form of a semantic commitment (in the case of $*$, the statement is global, i.e., universally agreed upon).

$\Diamond_s \phi$ (or $\Diamond_* \phi$) reads as “[From the point of view s,] in some sense ϕ”. In practice, this means that the standpoint s (or $*$) has no argument to rule out ϕ, and hence, it is acceptable to interpret s (or $*$) in such a way that ϕ holds.

In scenarios with multiple perspectives like Example 1, we can use standpoint logic to represent and infer different kinds of facts, namely (a) global or standpoint-relative statements, (b) unequivocal, and ‘*in some sense*’ statements, (c) hierarchies and globality of standpoints, such as $(IB \preceq HP)$, which models that the standpoint IB is ‘subsumed’ by the standpoint HP, and (d) ‘complex alignments’ between standpoints, such as $\Box_{HP}[\mathsf{Green}] \rightarrow (\Box_{CT}[\mathsf{Green}] \vee \Box_{HP}[\mathsf{Lime}])$. Intuitively the latter would mean that if Green holds according to HP, then either Green also holds according to CT or the colour is specifically Lime for HP (or both).

In the rest of this section, we demonstrate how to specify Example 1 using simple statements of the kinds (a) and (b), and we will present some of the inferences that we can obtain. Later, in Section 5, we will address more complex statements of the kinds (c) and

(d) by extending and enriching the same example, after presenting the formal specification of the language in Section 3. We now proceed with a simple standpoint formalisation of our example:

(1). $\square_{*}[(\mathsf{Yellow} \vee \mathsf{Green} \vee \mathsf{Lime}) \rightarrow \mathsf{Colour}]$
(2). $\square_{CT}[\mathsf{Colour} \leftrightarrow (\mathsf{WC} \vee \mathsf{CC})]$
(3). $\square_{CT}[(\mathsf{CC} \leftrightarrow \neg\mathsf{WC})]$
(4). $\square_{CT}[((\mathsf{Green} \vee \mathsf{Lime}) \rightarrow \mathsf{CC})) \wedge (\mathsf{Yellow} \rightarrow \mathsf{WC})]$
(5). $\square_{HP}[\mathsf{Lime} \rightarrow (\mathsf{Yellow} \wedge \mathsf{Green})]$

From this, we can derive the expected facts for each theory/standpoint specifically, such as ‘Unequivocally, according to CT, if a colour is lime then it is not yellow’ ($\square_{CT}[(\mathsf{Lime} \rightarrow \neg\mathsf{Yellow}]$), and general findings, such as ‘Under some interpretations, colours can be classified into cold and warm’ ($\lozenge_{*}[\mathsf{Colour} \rightarrow (\mathsf{CC} \vee \mathsf{WC})]$).
The crucial aspect of this framework in contrast to the merging strategy is that the set of sentences (1)-(5) is not inconsistent, so all axioms and known alignments can be jointly represented. Standpoint logic escapes global inconsistency, without removing alignments or relations and avoiding the duplication of entities, because the model theory based on Kripke-structures (cf. Section 3) forces consistency only within standpoints and precisifications, but allows for the specification of sets of standpoints that are inconsistent among them, which are modelled as disjoint sets. With this, we overcome the *merge* tradeoff discussed in Section 1.

3. Syntax and Semantics of Standpoint Logic

We now formally specify propositional *Standpoint Logic*, denoted with $\mathbb{S}$. Essentially, we define it as a multi-modal normal logic satisfying **KD45** (and **S5** in the case of the universal standpoint),[3] and additionally the stronger axioms **4’**, **5’** and **P**, describing the interaction of different standpoints. Hence, the formalism is an extension of well-known systems used for epistemic logics. We first define the syntax of $\mathbb{S}$.

Definition 1 (Syntax of Propositional Standpoint Logic). *A* vocabulary *is a tuple of the form $\mathcal{V} = \langle \mathcal{P}, \mathcal{S} \rangle$, where $\mathcal{P}$ is a non-empty set of propositional variables and $\mathcal{S}$ is a non-empty set of standpoint symbols, containing the distinguished symbol $*$. We denote elements of $\mathcal{P}$ by p and elements of $\mathcal{S}$ by s, potentially with extra decorations. The language $\mathcal{L}_{\mathbb{S}}$ of* $\mathbb{S}$ Propositions *(denoted by ϕ, ϕ_1, ϕ_2) are defined by*

$$\phi ::= s' \preceq s \mid p \mid \neg\phi \mid (\phi_1 \wedge \phi_2) \mid \square_s \phi.$$

where $s', s \in \mathcal{S}$, $p \in \mathcal{P}$, and $\square_s$ is called the standpoint operator *for s. We call (sub)formulae of the form $s' \preceq s$* sharpening statements *while those of the form p are referred to as* atomic *propositions. We also allow the connectives $\vee$, $\rightarrow$, $\leftrightarrow$, and $\lozenge_s$ as shorthands with their usual definitions.*

---

[3] These are systems of modal logic characterised by the axioms in Table 1. The system **KD45** is characterised by the axioms in its name and **S5** by K, D, T, 4 and 5.

Axiom	Schema	Property	Axiom	Schema	Property
K	$\Box_s(\phi \to \psi) \to (\Box_s \phi \to \Box_s \psi)$	All	**4’**	$\Box_s \phi \to \Box_{s'} \Box_s \phi$	Trans-transitive
D	$\Box_s \phi \to \Diamond_s \phi$	Serial	**5’**	$\Diamond_s \phi \to \Box_{s'} \Diamond_s \phi$	Trans-Euclidean
4	$\Box_s \phi \to \Box_s \Box_s \phi$	Transitive	**T***	$\Box_* \phi \to \phi$	Reflexive
5	$\Diamond_s \phi \to \Box_s \Diamond_s \phi$	Euclidean	**P**	$(s' \preceq s) \to (\Box_s \phi \to \Box_{s'} \phi)$	

Table 1. Correspondence between the axioms in $\mathbb{S}$ and the properties of the relations of the models $\mathfrak{M}_{\mathbb{S}}$.

In addition, we can define other useful modal operators in $\mathcal{L}_{\mathbb{S}}$:

$$\mathcal{I}_s \phi := (\Diamond_s \phi \wedge \Diamond_s \neg\phi) \qquad \mathcal{D}_s \phi := (\Box_s \phi \vee \Box_s \neg\phi) = \neg\mathcal{I}_s \phi$$

$\mathcal{I}_s \phi$ expresses that the truth of ϕ is indeterminate (i.e. borderline) according to standpoint s, while $\mathcal{D}_s \phi$ expresses that it is determinate.

Semantics of modal logic are typically provided proof-theoretically, and in this spirit we now give a Hilbert-style axiomatic proof system for $\mathbb{S}$, before complementing it with a fitting model-theoretic semantics. We write $\vdash_{\mathbb{S}} \phi$ to mean that ϕ is a derivable theorem of $\mathbb{S}$ (i.e. ϕ is derivable not requiring any premises). As standpoint logic is built upon an underlying classical logic, all classically valid propositional formulas are theorems. In addition, the proof system of $\mathbb{S}$ provides the axiom schemas **K**, **D**, **T***, **4’**, **5’**, and **P** as displayed in Table 1. The inference rules of $\mathbb{S}$ are the standard ones for modal logic: classical theorems are provable (**RC**), all instances of the axioms are provable (**RA**), the classical *modus ponens* (**MP**: if $\vdash_{\mathbb{S}} \phi$ and $\vdash_{\mathbb{S}} \phi \to \psi$, then $\vdash_{\mathbb{S}} \psi$), and the rule of necessitation (**RN**: if $\vdash_{\mathbb{S}} \phi$, then $\vdash_{\mathbb{S}} \Box_s \phi$, for all standpoints $s \in S$).

We proceed with a corresponding model-theoretic semantics of $\mathbb{S}$. We first propose a definition that captures the notion of *standpoint* well but deviates from the usual definition via Kripke models. Thereafter, we will justify that choice by arguing that there is a one-to-one correspondence between models according to our semantics and a particular class of Kripke models, allowing us to relate our work to other frameworks and results.

Definition 2 (Semantics of Standpoint Logic)**.** *Given a vocabulary $\mathcal{V} = \langle \mathcal{P}, \mathcal{S} \rangle$, a model $\mathcal{M}$ (over $\mathcal{V}$) is a triple $\langle \Pi, \sigma, \delta \rangle$, where Π is a non-empty set of* precisifications*, while $\sigma : \mathcal{S} \to 2^{\Pi}$ and $\delta : \mathcal{P} \to 2^{\Pi}$ are functions that assign sets of precisifications to standpoint symbols and propositional variables, respectively, satisfying $\sigma(s) \neq \emptyset$ for all $s \in \mathcal{S}$ as well as $\sigma(*) = \Pi$. The set of all such models is denoted by $\mathfrak{M}_{\mathbb{S}}$. For a model $\mathcal{M}$ with a distinguished precisification $\pi \in \Pi$ we define satisfaction of formulae as follows:*

- $(\mathcal{M}, \pi) \vDash p$ *iff* $\pi \in \delta(p)$,
- $(\mathcal{M}, \pi) \vDash \neg\phi$ *iff* $(\mathcal{M}, \pi) \nvDash \phi$,
- $(\mathcal{M}, \pi) \vDash \phi_1 \wedge \phi_2$ *iff* $(\mathcal{M}, \pi) \vDash \phi_1$ *and* $(\mathcal{M}, \pi) \vDash \phi_2$,
- $(\mathcal{M}, \pi) \vDash \Box_s \phi$ *iff* $(\mathcal{M}, \pi') \vDash \phi$ *for all* $\pi' \in \sigma(s)$,
- $(\mathcal{M}, \pi) \vDash s' \preceq s$ *iff* $\sigma(s') \subseteq \sigma(s)$.

We read $(\mathcal{M}, \pi) \vDash \phi$ as: ϕ is true at the precisification π in model $\mathcal{M}$. We write $\mathcal{M} \vDash \phi$ (read: $\mathcal{M}$ is a model of ϕ) if $(\mathcal{M}, \pi) \vDash \phi$ holds for all $\pi \in \Pi$. We call ϕ satisfiable *if it has a model and* valid *if every element of $\mathfrak{M}_{\mathbb{S}}$ is a model of it.*

As immediate consequence from this definition, we obtain that $\mathcal{M} \vDash \phi$ if and only if $\mathcal{M} \vDash \Box_* \phi$. Moreover, ϕ is valid if and only if $\Diamond_* \neg\phi$ is unsatisfiable.

$\mathfrak{M}_{\mathbb{S}}$ is not defined in the usual Kripke-style way using accessibility relations. Rather, the latter are replaced by the function σ following the intuition of *standpoint*, where standpoint symbols are associated with non-empty sets of precisifications.
However, **Definition 2** can be easily recast in terms of a set of accessibility relations $\mathcal{R} = \{R_s \mid s \in \mathcal{S}\}$ over the set of precisifications Π (as in standard Kripke semantics), by letting

$$R_s := \Pi \times \sigma(s) = \{\langle \pi, \pi' \rangle \mid \ \pi \in \Pi, \pi' \in \sigma(s)\}$$

That is, for each standpoint s, the relations thus obtained connect *all* precisifications with all those in $\sigma(s)$. Then, we obtain the desired correspondence.

Lemma 1. $(\mathcal{M}, \pi) \vDash \Box_s \phi$ *if and only if* $(\mathcal{M}, \pi') \vDash \phi$ *for all* π' *with* $(\pi, \pi') \in R_s$.

Soundness and completeness of the presented proof system with respect to the model-theoretic semantics can be shown by standard arguments. This includes relating the properties of our models to the axiomatisation of $\mathbb{S}$ using the well understood correspondence theory [21] (cf. Table 1). In particular, by construction, the standpoint accessibility relations are serial, transitive and euclidean, and the standpoint $*$ is a universal relation. This indeed corresponds to the **KD45** axiomatisation for the standpoints and the **S5** for $*$.

Discussion. As we have already mentioned, the main formal particularities of *standpoint logic* with regards to other modal frameworks are the axioms **P**, **4'** and **5'**. Axiom **P** captures the meaning of $\preceq$, by ensuring that any proposition considered definite in a given standpoint is also considered definite in any sharper standpoint. This simple mechanism is the basis for the construction of hierarchies of standpoints as well as for combinations. The axioms **4'** and **5'** are the interaction axioms, and are stronger than the well known modal axioms **4** and **5** (which are immediately derivable). This means that assertions such as $(\Box_a \Diamond_b \Box_c \ldots \Box_s)\phi$ can be simplified into $\Box_s \phi$. While this may seem unrealistic at first glance, and only motivated by the reduction of complexity it brings about, it is in fact a desirable feature. The key is that standpoints do not model the epistemic state of agents (in which case an assertion like 'Agent α knows that agent β knows ϕ' makes sense), they model the set of semantic commitments associated to a particular perspective. Moreover, we note that standpoints are not allowed to be empty (by axiom **D**), for they pick up compatible precisifications; if there are none, then the standpoint is deemed incoherent.

4. Translation into One-Variable First-Order Logic and Complexity

We next turn to the question regarding the difficulty of reasoning in $\mathbb{S}$. To this end, we will provide a polytime translation from $\mathbb{S}$ into one-variable first-order logic, which not only settles the above question but is also interesting in its own right.

Definition 3. *The function* trans : $\mathcal{L}_{\mathbb{S}} \rightarrow \mathcal{L}_{FO1}$, *mapping* $\mathbb{S}$ *formulae to formulae in one-variable first-order predicate logic, is recursively defined as follows (with symbols from* $\mathcal{P}$ *and* $\mathcal{S}$ *repurposed as unary predicates,* $s', s \in \mathcal{S}$ *and* $p \in \mathcal{P}$*):*

$$\begin{aligned} \mathrm{trans}(p) &= p(x) & \mathrm{trans}(\phi_1 \wedge \phi_2) &= \mathrm{trans}(\phi_1) \wedge \mathrm{trans}(\phi_2) \\ \mathrm{trans}(\neg\phi) &= \neg\mathrm{trans}(\phi) & \mathrm{trans}(\Box_s \phi) &= \forall x.(s(x) \rightarrow \mathrm{trans}(\phi)) \\ & & \mathrm{trans}(s' \preceq s) &= \forall x.(s'(x) \rightarrow s(x)) \end{aligned}$$

Finally, for an $\mathbb{S}$ *formula* ϕ *with occurrences of standpoint constants* $s_1, \ldots, s_k$*, let*

$$\mathrm{Trans}(\phi) := \forall x.(\mathrm{trans}(\phi)) \wedge \forall x.(*(x)) \wedge \exists x.(s_1(x)) \wedge \ldots \wedge \exists x.(s_k(x)).$$

We note that Trans produces formulae of linear size w.r.t. $|\phi|$. We now show that the translation is indeed satisfiability preserving, as intended.

Theorem 2. *An* $\mathbb{S}$ *formula* ϕ *is satisfiable if and only if* $\mathrm{Trans}(\phi)$ *is FO-satisfiable.*

Proof. ($\Rightarrow$) Assuming satisfiability ϕ consider some model $\mathcal{M} = \langle \Pi, \sigma, \delta \rangle$ of it. Let now $\mathrm{FO}(\mathcal{M})$ denote the first-order interpretation with domain Π, satisfying $s^{\mathrm{FO}(\mathcal{M})} = \sigma(s)$ for all $s \in \mathcal{S}$ and $p^{\mathrm{FO}(\mathcal{M})} = \delta(p)$. Then it can be shown by a straightforward structural induction over ϕ that, for every $\pi \in \Pi$, $(\mathcal{M}, \pi) \models \phi$ if and only $\mathrm{FO}(\mathcal{M}), \{x \mapsto \pi\} \models \mathrm{trans}(\phi)$. It is then a direct consequence, that $\mathrm{FO}(\mathcal{M})$ is a model of the first conjunct of $\mathrm{Trans}(\phi)$. Satisfaction of the other conjuncts follows from **Def. 2** via the definition of $\mathrm{FO}(\mathcal{M})$. Hence $\mathrm{FO}(\mathcal{M})$ is a model of $\mathrm{Trans}(\phi)$, witnessing its satisfiability.
($\Leftarrow$) Consider a first-order model $\mathcal{M}'$ of $\mathrm{Trans}(\phi)$. We now define the $\mathbb{S}$ model $\mathbb{S}(\mathcal{M}') = \langle \Pi, \sigma, \delta \rangle$ by letting Π be the domain of $\mathcal{M}'$ and stipulating $\delta(p) = p^{\mathcal{M}'}$ for all $p \in \mathcal{P}$ as well as $\sigma(s) = s^{\mathcal{M}'}$ whenever $s \in \{s_1, \ldots, s_k\}$ and $\sigma(s) = \Pi$ otherwise. It is easily checked that, due to the second to last conjunct of Trans, the sturcture thus defined is indeed an $\mathbb{S}$ model. In order to show that $\mathbb{S}(\mathcal{M}')$ is a model of ϕ, we can proceed as before and show by a straightforward structural induction over ϕ that, for every $\pi \in \Pi$, $(\mathbb{S}(\mathcal{M}'), \pi) \models \phi$ if and only $\mathcal{M}', \{x \mapsto \pi\} \models \mathrm{trans}(\phi)$. Thus, the established modelhood of $\mathbb{S}(\mathcal{M}')$ ensures satisfiability of ϕ. ☐

It is folklore that the satisfiability problem of one-variable first-order logic is decidable and, in fact, in NP, membership even having been established for much more expressive logics [22]. As Trans realizes a polynomial reduction from satisfiability in $\mathbb{S}$ to satisfiability to one-variable first-order logic, this membership carries over. On the other hand, $\mathbb{S}$ subsumes propositional logic, which is known to have an NP-hard satisfiability problem. Hence we can conclude the following.

Corollary 3. *The satisfiability problem of* $\mathbb{S}$ *is* NP*-complete.*

5. Integrating Different Perspectives with SL

So far, we have shown how we can formalise the conflicting perspectives of our Example 1 to overcome some limitations of traditional merging approaches. We now extend this example to briefly illustrate other capacities of the language, covering the representation of alignments, standpoint nesting and standpoint combination. When representing alignments, we will assume that the alignments themselves have been obtained via traditional ontology matching techniques or that they may be known to the domain experts.

'Alignment' representation. Standpoint logic allows for a reasonable nuance in the representation of correspondences between the entities of different standpoints compared to other formalisms. On the one hand, we usually interpret simple scenarios where we have

an equivalence relation like $(\equiv, 0.78)$[4] as both entities being different perspectives of a single concept. This is the case that we have already widely covered, and it only involves, when necessary, the uniform renaming of entities.
The case of subsumed relations is more nuanced. What are called subsumptions in the context of knowledge integration can be either 'genuine subsumptions', such as $\langle O1{:}\mathit{Lime},\ O2{:}\mathit{Colour},\ \subseteq,\ 0.84\rangle$ or 'sharpenings', where the concept is intuitively equivalent but one ontology has stricter criteria of application than the other, such as $\langle O1{:}\mathit{Green},\ O2{:}\mathit{Green},\ \subseteq,\ 0.73\rangle$. Following the reported evidence in [11], we assume that these cases can be often recognised because an additional equivalence alignment is also found, $\langle O1{:}\mathit{Green},\ O2{:}\mathit{Green},\ \equiv,\ 0.56\rangle$, as illustrated in Figure 1. In this case, we suggest that it is more appropriate to understand the relation as a subsumption between the standpoints on that concept, rather than a subsumption of different concepts:

(6). $\Box_{s_1}[\mathsf{Green}] \rightarrow \Box_{s_2}[\mathsf{Green}]$

Beyond the more faithful representation of the correspondence, this approach limits the multiplication of entities in the merged ontology, which can otherwise hinder its usability. Finally, the framework allows for the representation of correspondences as complex as allowed by the base logic language. E.g., $\Box_{HP}[\mathsf{Green}] \leftrightarrow (\Box_{CT}[\mathsf{Green}] \vee \Box_{HP}[\mathsf{Lime}])$ specifies that if Green holds according to *HP*, then either Green also holds according to *CT* or the colour is specifically Lime for *HP* (or both). Note that the behaviour of these sentences is in fact more similar to *bridge rules*, because rather than unifying the theories, they only establish a correspondence between what holds for different standpoints.

Standpoint Hierarchies and Combinations. Statements like (6) are radically different from those of the kind $(s_2 \preceq s_1)$, as the latter establishes a relation between full standpoints. Let us consider a use case scenario of $\preceq$,

Example 2. An Ink brand with the standpoint s_{IB} reuses the categorisation of *HP*, and in addition it specifies that Ochre and Gold are (types of) Yellow.

(7). $\Box_{IB}[(\mathsf{Gold} \vee \mathsf{Ochre}) \rightarrow \mathsf{Yellow}]$
(8). $IB \preceq HP$

Statement (8) specifies that *IB* is a sharpening of *HP*, that is, *IB* satisfies all the constraints of *HP* as well as its own commitments (7). Semantically, this is modelled as a subset relationship: the precisifications belonging to *IB* are a subset of those belonging to *HP*. This is relatable to the process of *importing* an ontology. Yet, sharpening a standpoint is slightly different from importing it: From the formulae (7) and (8), we can infer not only propositions that hold according to *IB* but also about *HP*. For instance, the two statements imply that, under *HP*'s perspective, it is admissible to interpret ochre as a kind of yellow: $\Diamond_{HP}[\mathsf{Ochre} \rightarrow \mathsf{Yellow}]$. As a consequence, if there was another standpoint *IB2* such that $\Box_{IB2}[\mathsf{Ochre} \rightarrow \neg\mathsf{Yellow}] \wedge IB2 \preceq HP$, and we knew $\Box_{HP}[\mathsf{Ochre}]$, then one could infer that $\mathcal{I}_{HP}[\mathsf{Yellow}]$, that is, that yellow is inherently borderline or indeterminate for s_{HP} since we have evidence that it can be sharpened in opposing ways.

---

[4] We use here the standard notation in ontology alignment, where matchings between two entities have a relation type, in this case $\equiv$, and a similarity or confidence measure, in this case 0.78.

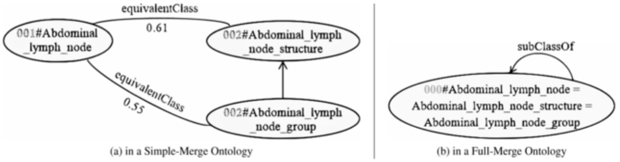


Figure 3. Ambiguous equivalence correspondences leading to redundancies and cycles. Reprinted from [11].

With regards to the combination of standpoints, let us consider an example of reasoning in scenarios where two independently developed models are relevant, namely our previous *IB* and another standpoint *PB*.

Example 3. Assume there is a paint brand with standpoint *PB*, coming with some more definitions and axioms. If the standpoints *IB* and *PB* overlap, we can introduce a new, joint standpoint *IBPB* and define it as sharpening of both *IB* and *PB* to merge the theories.

(9). $IBPB \preceq IB \ \wedge \ IBPB \preceq PB$

Note that, so far, our framework does not allow to precisely refer to the intersection of the two standpoints (only at any possible subset of their intersection), but we will discuss an extension to this effect in the conclusion.

6. Application in the Biological Domain

In this section we consider two possible applications of the standpoint framework in the context of biological ontologies. First, it might serve as an alternative approach to reported integration challenges that arose in the *LargeBio* track of the Ontology Alignment Evaluation Initiative 2020 [23]. Second, it may help to address the semantic heterogeneity challenges around the concept *forest* in the EnvO ontology.

First, let us consider an alignment scenario extracted from an experiment in [11], where the goal is the holistic integration of three ontologies (using pairwise alignments) from the *LargeBio* track: FMA (Foundational Model of Anatomy), SNOMED-CT (Clinical Terms), and NCI (National Cancer Institute Thesaurus). In what follows, we will refer to the entities of these sources by their initials, for the sake of brevity.

One case of this scenario (extracted from [11]) is illustrated in Figure 3, which contains a snippet from the alignments found between FMA and SNOMED-CT on the left, and their naive integration on the right. The integration displays subsumption redundancies and a

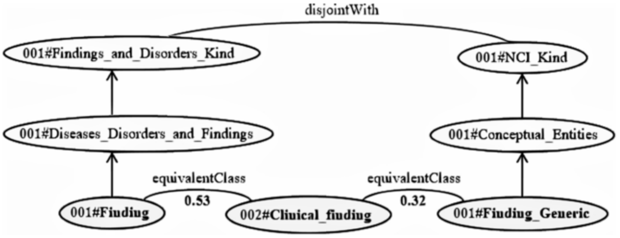


Figure 4. Ambiguous equivalence correspondences leading to inconsistency. Reprinted from [11].

subsumption cycle formed because of the addition of two equivalence correspondences having the same source class '001#Abdominal_lymph_node' (ALN), resulting in three equivalent classes merged together to constitute a single class. In [11], it is proposed to fix this merge by giving up one of the alignments, hence concluding either $\mathsf{ALN} \equiv \mathsf{ALNG}$ or $\mathsf{ALN} \equiv \mathsf{ALNS}$.

The Standpoint Logic Approach. Let us consider that the output of the merge process is represented under a standpoint (#001/2), that contains assertions relevant to all the statements that are not flagged as problematic. Then, we can merely state (10) and optionally (11), to strengthen the axiomatisation of the combination.

(10). $(\#001) \preceq (\#001/2) \wedge (\#002) \preceq (\#001/2)$
(11). $\square_{(\#001/2)}[(\mathsf{ALN} \leftrightarrow \mathsf{ALNS}) \ \vee \ (\mathsf{ALNG} \leftrightarrow \mathsf{ALN})]$

The same approach can be taken to address more problematic scenarios that provoke inconsistency instead of 'malformation', such as the one displayed in Figure 4. In this case, the traditional approaches are (1) to give up on one of the alignments and (2) to remove the disjointness restriction. With standpoint logic, we can model this again as a structure of three standpoints or, alternatively, we can add complex alignments directly between the initial ontologies, such as $\square_{(\#002)}[CF] \leftrightarrow \square_{(\#001)}[F \vee FG]$.

Let us now consider the second application scenario: that of a standpoint ontology by design. In the process of designing a general-purpose ontology, addressing the semantic heterogeneity of some terms is often challenging. For instance, the need for adding several forest characterisations in the Environment Ontology (*EnvO*) was reported [24]. In the absence of frameworks supporting 'characterisations', big ontologies seeking generality rely on (i) weakening the precision (e.g. EnvO mostly uses *part_of* and *is_a* roles, and avoids disjointness), and (ii) formalising different (or very similar) overlapping entities, that correspond to different standpoints on a concept and lead to convoluted taxonomies.

The scenario in EnvO is as follows: as of October 2019, forests in *EnvO* are represented via two main classes, namely *forested area* and *forest ecosystem*, both of which have the same textual description. *Forested area* has 'forest' as a related synonym and links to the forest entry of Wikipedia, among other database cross-references. It is hence the 'de facto' concept for forest. *Forest ecosystem* and *forest biome* are intended to characterise the forest as an ecosystem and as a biome respectively. Additional classes subsumed by *vegetated area* also refer to forests (e.g. *area of evergreen forest*), yet they are not related to any of the main forest concepts, possibly in order to avoid conflicts. Figure 5 is an overview of the most important 'forest' entities and some of the main superclasses.

Our proposal is that designing ontologies such as EnvO in a modular way by means of standpoints may help in overcoming the tradeoff between generality, convolution and

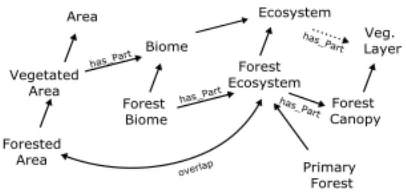


Figure 5. Fragment of the EnvO Ontology (2019).

precision. While radically redesigning EnvO forests based on thematic standpoints goes beyond the scope of this paper, we discuss one of many small details:
In Figure 5 we see that Forest_Biome is related to Ecosystem via parthood through Forest_Ecosystem and via subsumption through Biome. Yet, it does not seem intended for a biome to be a forest ecosystem and have a part that is also a forest ecosystem. Thus we assume that this could be better represented with two alternative standpoints, where one models that Forest_Biome has a part Forest_Ecosystem and Biome is not an Environment and the other models that Forest_Biome is a Forest_Ecosystem, and Biome is an Environment. With this, we'd avoid misuses and we capture the intuition that, depending on how we interpret the polysemous word Biome, we can think of a Forest_Biome being a Forest_Ecosystem or the latter just being part of it.

7. Related Work and Discussion

The importance of handling the interpretation of information in relation to its standpoint or context, and of understanding relationships between those, has been recognised by many researchers in AI [25]. This has led to the proposal of a variety of systems of representation, in rather overlapping areas of research and with diverse nomenclatures.

Contextual Ontologies. Semantic variation is often associated with differences of *context*. Our approach has some similarities to context logics in the style proposed by McCarthy [25], such as the modal framework in [26]. However, this tradition focuses on modelling contexts, and treats them as full-fledged formal objects over which one can express first-order properties. In contrast, Standpoint Logic is more suitable for the many cases in which a detailed formalisation of the contexts involved in a domain is either unnecessary or unfeasible, or where the interest resides in the perspectives or standpoints themselves rather than the context in which they occur.
A contextual framework where contexts are mere labels is [27]. However, in Benslimane's framework, context-labels can only be used in rather restricted scenarios and contexts are formally closer to the *ontology viewpoints* than to the standpoints of our logic, leaving no room for standpoint relations and compositions, which are crucial in many application scenarios as seen in the examples (cf. Section 6).

Ontology Viewpoints. The notion of ontology views is inherited from the well-known view mechanism in database theory. Most research in this area follows that tradition, and focuses on the presentation of partial (and consistent) views (partitions) on the content of a single ontology, which may be interesting to different agents [28,29].
However, some works consider potentially conflicting viewpoints, such as [30] and the similar and more developed [31]. Both share our motivation, yet they approach it differently and tailor their framework to description logics, in a style similar to Benslimane's work. Hemam and Boufaida [31] define seven types of elements (viewpoints, global classes, local classes, global properties, local properties, bridge rules and individuals) in a nested structure, leading to a rather intricate logic, for which no complexity bounds are provided.
Instead of implementing the ad-hoc intuition of 'viewpoints', our work just extends the base language (in this paper: propositional logic) with modal operators and gives the resulting logic a Kripke semantics. This leads to a simpler, more recognisable and more

expressive framework, that allows for the unrestricted use of the modal operators in any kind of sentence and supports, for instance, hierarchies and combinations of standpoints, inferences of partial truths, the preservation of consistency with penumbral connections and inferences about the standpoints themselves. As for consistency, [31] develops a mechanism that works by bringing global facts down to the viewpoints, while the modal framework ensures consistency also in the inverse direction, allowing e.g. for disambiguation strategies. For instance, in our example, one can infer that $\Diamond_*[\mathsf{Lime} \rightarrow \mathsf{Yellow}]$ and that $\Diamond_*[\mathsf{Yellow} \rightarrow \mathsf{WC}]$ but not $\Diamond_*[\mathsf{Lime} \rightarrow \mathsf{WC}]$ because CT and HP are not guaranteed to be compatible standpoints (cf. Fig. 2).

DDL Bridge Rules and ε-connections. In the area of ontology modularity, different formalisms such as DDL bridge rules [32] and ε-connections [33] have been proposed to specify the interaction between independent knowledge sources. These can be related to the present framework in that they provide mechanisms to establish links between conceptual models, similar to the role of assertions involving several standpoints such as $\Box_{HP}[\mathsf{Green}] \leftrightarrow (\Box_{CT}[\mathsf{Green}] \vee \Box_{HP}[\mathsf{Lime}])$, yet the motivation is inherently different: while the standpoint framework focuses on integrating possibly overlapping knowledge into a global source, DDL Bridge rules and ε-connections have been proposed to connect standalone modules. Moreover, both have been proposed for DL languages.

In contrast to standpoint statements, DDL bridge rules [32] are directional (INTO or ONTO) relations between concepts of different modules, such that an INTO rule between a concept A in module M_1 and a concept B in module M_2, $i : A \xrightarrow{\sqsubseteq} j : B$, does not entail the converse ONTO rule $j : B \xrightarrow{\sqsupseteq} i : A$.

ε-connections are a combination method that takes the union of the combined modules M_1 and M_2, enriched with operators capable of talking about the link relations that the ε-connection establishes between them. This behaviour can be mimicked in a standpoint style by encapsulating the ε-connection into a standpoint s_ε that encodes those links and is subsumed by the standpoints s_{M_1} and s_{M_2}. ε-connections however require the vocabularies of the connected modules to be disjoint, which is reasonable in the context of ontology modularity but an important limitation for our main subject of interest: scenarios in which there are competing perspectives on the semantics of a shared vocabulary.

8. Conclusions and Future Work

The semantic heterogeneity of natural language together with the diversity of human world views are at the root of many knowledge interoperability scenarios. As an alternative to the mainstream unification strategy, this paper introduces a logic formalism based on the notion of standpoint that is suitable for knowledge representation and reasoning with sets of possibly conflicting perspectives or characterisations of a domain. We explore how different agents can establish their standpoints, which typically involves specifying constraints and relations (amounting to making ‘ontological commitments’) but not necessarily subscribing to a single sharp interpretation. Natural reasoning tasks over such multi-standpoint specifications include gathering unequivocal or undisputed knowledge, determining knowledge that is relative to a standpoint or a set of them, and contrasting the knowledge that can be inferred from different standpoints. The fact that the proposed formalism preserves the complexity of the propositional base language, having an NP-complete algorithm for satisfiability in contrast to the PSPACE-completeness of multi-modal epistemic logics in general, indicates that the framework can have interesting, technically feasible applications in areas such as ontology alignment and concept negotiation, and knowledge aggregation.

Moreover, in contrast to other proposals of multi-perspective frameworks (such as *ontology viewpoints*), our framework is rooted in a well established philosophical theory of language, supervaluationism, and thus can be linked to a theoretic body of work. In addition, the use of modalities makes the language easily recognisable for a broad community of researchers and practitioners, and allows for the expression of assertions that are only guaranteed to hold in some sense or that are borderline, which becomes useful in scenarios involving collections of interpretations.

There are several directions of future research. First, the set-theoretic structure of standpoints not only facilitates the establishment of hierarchies of interpretations, but it also makes it possible to define an algebraic calculus allowing to define complex standpoints out of atomic ones by means of union ($s_1 \cup s_2$, representing the integration of knowledge coming from two different sources), intersection ($s_1 \cap s_2$, collecting the "agreed-on knowledge" shared between two standpoints) or difference ($s_1 \setminus s_2$, representing a sharpening of standpoint s_1 through the exclusion of all precisifications pertaining to s_2). It is easy to see that an extension of our formalism by such standpoint–algebraic expressions comes at no additional cost in the NP complexity of reasoning, since the translation to one-variable first order logic presented in Section 4 can be easily adapted.

A shortcoming of our current work is that, as a starting point for our investigations, we have chosen to define the language for a propositional base, while today's knowledge representation formalisms use more expressive logics. Yet, because modal frameworks are well understood also for more advanced logics such as description logics and (fragments of) first-order logic, the corresponding adaptation of syntax and semantics should not be problematic and is object of current work. Of course, such extensions will also necessitate to determine the underlying complexities, to specify the corresponding proof-theoretic calculi and to develop strategies to employ off-the-shelf reasoners for standpoint-enhanced reasoning in practical scenarios. Given that the reason why the first order logic translation is well-behaved in terms of complexity is the small model property of the logic, we expect to maintain this good behaviour for more expressive base languages.

References

[1] Wittgenstein L. Philosophical Investigations. tr. G.E.M. Anscombe ed. Blackwell; 1958.

[2] Guarino N. Formal Ontology and Information Systems. In: Procs. of the 1st Int. Conf. on Formal Ontology in Information Systems. vol. 46. IOS Press; 1998. p. 3–15.

[3] Kalfoglou Y, Schorlemmer M. Ontology mapping: the state of the art. The knowledge engineering review. 2003;18(1):1–31.

[4] Euzenat J, Mocan A, Scharffe F. Ontology Alignments. In: Hepp M, Leenheer PD, Moor AD, Sure Y, editors. Ontology Management. Computing for Human Experience. vol. 7. Springer; 2008. p. 177–206.

[5] Otero-Cerdeira L, Rodríguez-Martínez FJ, Gómez-Rodríguez A. Ontology matching: A literature review. Expert Systems with Applications. 2015;42(2).

[6] Pesquita C, Faria D, Santos E, Couto FM. To repair or not to repair: Reconciling correctness and coherence in ontology reference alignments. In: Procs. of the 8th Int. Conf. on Ontology Matching. vol. 1111. CEUR; 2013. p. 13–24.

[7] Solimando A, Jiménez-Ruiz E, Guerrini G. Minimizing conservativity violations in ontology alignments. Knowledge and Information Systems. 2017;51(3).

[8] Jiménez-Ruiz E, Cuenca Grau B, Horrocks I, Berlanga R. Ontology integration using mappings: Towards getting the right logical consequences. In: Procs. of the 6th European Semantic Web Conf.. vol. 5554. Springer; 2009. p. 173–187.

[9] Megdiche I, Teste O, Trojahn C. An extensible linear approach for holistic ontology matching. In: Procs. of the 15th Int. Semantic Web Conf.. vol. 9981. Springer; 2016. .

[10] Thiéblin E, Haemmerlé O, Hernandez N, Trojahn C. Survey on complex ontology matching. Semantic Web. 2020;11(4).

[11] Osman I, Ben Yahia S, Diallo G. Ontology Integration: Approaches and Challenging Issues. Information Fusion. 2021;71:38–63.

[12] Bouquet P, Ghidini C, Giunchiglia F, Blanzieri E. Theories and uses of context in knowledge representation and reasoning. Journal of Pragmatics. 2003 3;35(3):455–484.

[13] Keefe R, Smith P. Theories of vagueness. In: Keefe R, Smith P, editors. Vagueness: A reader. The MIT press; 1997. p. 1–57.

[14] Mehlberg H. The Reach of Science. University of Toronto Press; 1958.

[15] Fine K. Vagueness, truth and logic. Synthese. 1975;30(3-4):265–300.

[16] Bennett B. Standpoint Semantics: a framework for formalising the variable meaning of vague terms. In: Cintula P, Fermüller C, Godo L, Hájek P, editors. Understanding Vagueness: Logical, Philosophical and Linguistic Perspectives. Studies in Logic. College Publications; 2011. p. 261–278.

[17] Gómez Álvarez L, Bennett B. Dealing with conceptual indeterminacy: A framework based on supervaluation semantics. In: Joint Procs. of MedRACER and WOMoCoE. CEUR; 2018. .

[18] Varzi AC. Supervaluationism and Its Logics. Mind. 2007;116(463):633–676.

[19] Williams JRG. Supervaluationism and Logical Revisionism. The Journal of philosophy. 2008;105(4):192–212.

[20] Cobreros P. Supervaluationism and classical logic. In: Int. Workshop on Vagueness in Communication. VIC 2009. Springer; 2011. p. 51–63.

[21] Lemmon EJ, Scott D. An introduction to modal logic. Oxford: Blackwell; 1979.

[22] Pratt-Hartmann I. On the Computational Complexity of the Numerically Definite Syllogistic and Related Logics. Bulletin of Symbolic Logic. 2008;14(1):1–28.

[23] Pour N, Algergawy A, Amini R, Faria D, Fundulaki I, Harrow I, et al. Results of the ontology alignment evaluation initiative 2020. In: Procs. of the 15th Int. Workshop on Ontology Matching. vol. 2788. CEUR; 2020. p. 92–138.

[24] Buttigieg PL, Pafilis E, Lewis SE, Schildhauer MP, Walls RL. The environment ontology in 2016: bridging domains with increased scope, semantic density, and interoperation. Journal of Biomedical Semantics. 2016 12;7(1):57.

[25] McCarthy J, Buvac S. Formalizing context (expanded notes). 1997.

[26] Klarman S, Gutiérrez-Basulto V. Description logics of context. Journal of Logic and Computation. 2013;26(3):817–854.

[27] Benslimane D, Arara A, Falquet G, Maamar Z, Thiran P, Gargouri F. Contextual ontologies motivations, challenges, and solutions. In: Procs. of the 4th Int. Conf. on Advances in Information Systems. Springer; 2006. p. 168–176.

[28] Rajugan R, Chang E, Dillon TS. Ontology views: A theoretical perspective. In: On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. vol. 4278 LNCS - II. Springer; 2006. p. 1814–1824.

[29] Bolchini C, Curino C, Quintarelli E, Schreiber F, Tanca L. Context information for knowledge reshaping. Int Journal of Web Engineering and Technology. 2009;5(1):88–103.

[30] Ribière M, Dieng-Kuntz R. A viewpoint model for cooperative building of an Ontology. In: Procs. of the 10th Int. Conf. on Conceptual Structures. Springer; 2002. p. 220–234.

[31] Hemam M, Boufaïda Z. MVP-OWL: A multi-viewpoints ontology language for the Semantic Web. Int Journal of Reasoning-based Intelligent Systems. 2011;3(3-4):147–155.

[32] Borgida A, Serafini L. Distributed Description Logics: Assimilating information from peer sources. Journal on Data Semantics. 2003;2800:153–184.

[33] Kutz O, Lutz C, Wolter F, Zakharyaschev M. E-connections of Description Logics. Description Logics Workshop. 2003;81.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210368

Towards Formalisation of Concept Descriptions and Constraints

Matt SELWAY [1], Markus STUMPTNER and Wolfgang MAYER
Industrial AI Research Centre, UniSA STEM,
University of South Australia, Australia

Abstract. The integrated management of industrial systems in future environments like Industry 4.0 requires the effective management of information throughout the engineering life cycle. As systems pass through phases of design, construction, operation, maintenance, renewal or replacement, they will be administered via different information ecosystems, requiring changing perspectives on their descriptive information. A central role in the interplay of software and hardware artefacts, functions, documentation and managing software is played by the *description*s of concepts (i.e. formalised definitions of concepts within the domain of quantification). In this paper we propose a unified formalisation of *description*s that permits consistent analysis of the relationships between the designs, types, products, and concrete artefacts that can be found in the industrial engineering life-cycle. The approach is consistent with our earlier framework that describes artefacts, requirements and functional roles in the context of the DOLCE foundational ontology.

Keywords. Interoperability, Artefacts, Descriptions, Relationships

1. Introduction

Effective construction and operation of industrial plants requires the management of extensive bodies of information about the plant and its surrounding operational and maintenance activities across the entire life cycle of the plant. Figure 1 depicts the life cycle of a particular “object” in engineering parlance, such as a pump that is part of a vehicle or an industrial plant. The pump itself is a complex object composed from multiple parts—with its specification, design information, and maintenance/fault records reflecting this subdivision—while being itself only a small part of the whole plant. Each stage, from requirements through specification, design, construction, operation, and maintenance, may be subject to information being handled by different software systems based on different languages, data models, or assumptions concerning scope and interpretation of data. As a result, establishing and sustaining interoperability between these heterogeneous systems is a long standing challenge, in particular in heavy industry sectors.

---

[1] Corresponding Author: Matt Selway, University of South Australia, GPO Box 2471, Adelaide SA 5001, Australia; E-mail: `matt.selway@unisa.edu.au`.

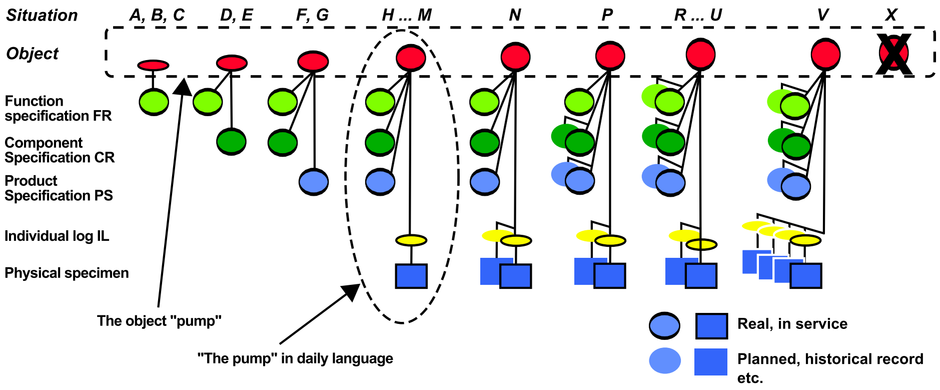


Figure 1. Engineering Life Cycle of a Pump Object (Source: POSC Caesar Association)

Information modelling and exchange standards have been developed to facilitate sharing of information among these information systems, but they remain generally confined to particular subsections of the life cycle. Ontology based information modelling offers to help overcome the information silos by providing a principled view of the problem domain and the associated information artefacts. Such an overarching reference model can then be used to mediate between the individual system's information models by expressing each individual information system in terms of the reference model and exploiting the resulting mappings in the transformations that implement the information exchange protocols. For the construction of suitable transformations and to maintain consistent information across system boundaries, automated validation and integration of information are key. In [1], we presented a framework for such a reference model based on explicit ontologically sound representation of artefacts.

This paper extends the formal coverage of the framework to manage, apply and validate machine interpretable content of the specifications and descriptions associated with an artefact in different life cycle stages, without the development of custom application code for each aspect. In Section 2, we examine the basic requirements for such an explicit representation and introduce declarative *constraints* that embody domain specific invariants on the instances of the ontology that represent a particular collection of technical artefacts relevant to an application. We characterise the properties of constraints and distinguish necessary and obligatory constraints in the context of evolving information. In Section 3 we validate the formal model by applying it to describe the SLICER approach [2] for representation of industrial artefact domains, showing that the framework can restate the core axioms of SLICER within the extended DOLCE ontology from [1].

2. Basic Formulation

In this section we formalise the foundations of our approach. As in [1], the representation is again formalised as extensions to the DOLCE foundational ontology [3] enriched with the notion of *social concept* and *description* introduced in [4], relationships reified in the domain of quantification suggested in [5], artefacts

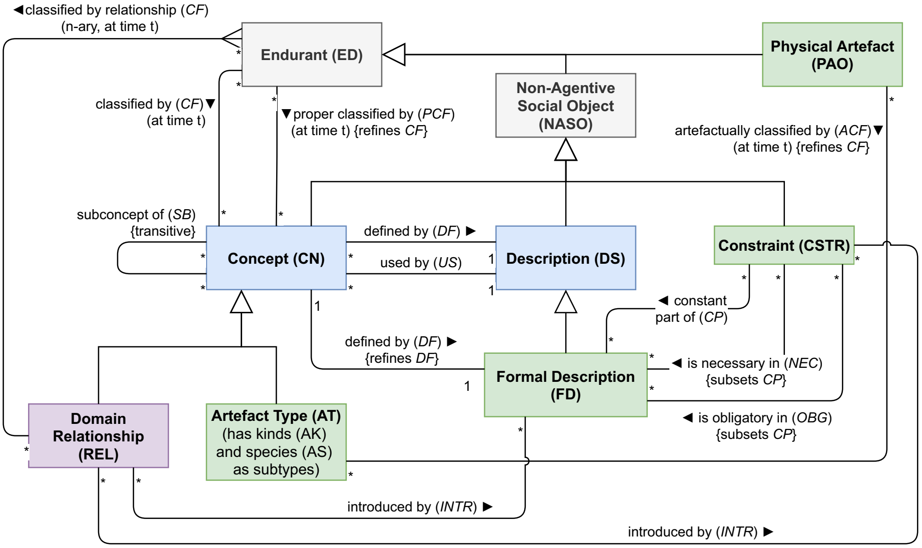


Figure 2. Conceptual overview of the types and relations in the formalisation

proposed in [6], and functions and roles discussed in [7]. For brevity, we consider only a subset of categories to do with concepts and their descriptions. An overview of these concepts is illustrated in Figure 2. In particular, we focus on artefactual objects and their specifications.

(1) $\mathsf{CN}(x) \rightarrow \mathsf{NASO}(x)$ (Concepts are non-agentive social objects)
(2) $\mathsf{REL}(x) \rightarrow \mathsf{CN}(x)$ (Domain Relationships (types) are concepts)
(3) $\mathsf{AT}(x) \rightarrow \mathsf{CN}(x)$ (Artefact Types are concepts)
(4) $\mathsf{AT}(x) \rightarrow \mathsf{AK}(x) \vee \mathsf{AS}(x)$ (Artefact types are A.Kinds or A. Species)[2]

It should be noted that since the basis of our formulation reifies concepts and their descriptions in the domain of quantification, so too must the constraints be reified within the domain of quantification. Therefore, the typical approach that would use, say, modal logic and axioms at the top-level axiomatisation is inappropriate for defining the domain specific constraints. Moreover, the approach taken here, which avoids the use of modal logic, quarantines the domain specific definitions and constraints from the framework used to describe them, allowing local re-evaluation of constraints when data changes in the system (e.g., a new temperature value is sensed). This is important in an information system that may be managing very large sets of instances and constant data updates.

Descriptions (really, *formal* descriptions) uniquely define a concept and comprise constraints (CSTR), which can be *necessary* or *obligatory* (i.e., constraints that can be falsified without impacting an object's classification). This specialises the definition of [4] which allows a description to define multiple concepts or none at all.[3] We take an intensional view that requires concepts to be defined by (DF) a

---

[2] Artefact Kinds are more general types such as 'Pump', while Artefact Species are more detailed, physically and functionally, such as the C12 pump model sold by a manufacturer.

[3] This difference is due to formalising the content of *formal descriptions* to be *constraints* rather than other, simpler descriptions that do not define a concept at all; this does not change

description, indeed it is part of a concept's identity. The constraints extend the content of descriptions to more than the 'used by' relation of [4]. We maintain consistency with [4] in that we consider the semantic content of a description to not change over time (12, 13).

(5) $\mathsf{DS}(x) \rightarrow \mathsf{NASO}(x)$ (Descriptions are non-agentive social objects [4])
(6) $FD(x, y) \rightarrow \mathsf{DS}$ (Formal descriptions are descriptions)
(7) $DF(x, y) \rightarrow \mathsf{CN}(x) \wedge \mathsf{DS}(y)$ (The concept x is defined by the description y)
(8) $\mathsf{FD}(x) \rightarrow \exists! y\, \mathsf{CN}(y) \wedge DF(y, x)$ (9) $\mathsf{CN}(x) \rightarrow \exists! y\, \mathsf{DS}(y) \wedge DF(x, y)$
(10) $\mathsf{CSTR}(x) \rightarrow \mathsf{NASO}(x)$ (A constraint is a non-agentive social object)
(11) $P(x, y, t) \rightarrow \mathsf{ED}(x) \wedge \mathsf{ED}(y) \wedge \mathsf{T}(t)$ (Endurant x is a part of y during t [3])
(**d**1) $CP(x, y) \triangleq \exists t\, (PRE(y, t)) \wedge \forall t\, (PRE(y, t) \rightarrow P(x, y, t))$ (Constant Part [3])
(12) $NEC(x, y) \rightarrow \mathsf{CSTR}(x) \wedge \mathsf{FD}(y) \wedge CP(x, y)$ (x is necessary in y)
(13) $OBG(x, y) \rightarrow \mathsf{CSTR}(x) \wedge \mathsf{FD}(y) \wedge CP(x, y)$ (x is obligatory in y)
(14) $\mathsf{CSTR}(x) \wedge \mathsf{FD}(y) \wedge P(x, y) \rightarrow NEC(x, y) \vee OBG(x, y)$

Classification Necessary constraints must always evaluate to true for an object to be considered to be classified by the concept with the constraint. In contrast, obligatory constraints may be violated while still considering the object to be classified by the concept. To handle this, we weaken the definition of classification (CF) from [4] and introduce two specific types of classification: artefactual classification, ACF, and proper classification, PCF.

Artefactual Classification embodies the idea that an artefact is intended to be of a certain type, or is created to conform to a particular specification or artefact species [6]. However, it may not always conform to it, for example, if it is broken down. *Proper* classification, on the other hand, assumes the notion of full conformance to the constraints of the description and applies to non-artefactual concepts, e.g., social concepts, and roles. Importantly, proper classification allows for a concept to be classified by another concept, allowing the construction of multi-level hierarchies describing concepts at different levels of abstraction.

(15) $CF(x, y, t) \rightarrow \mathsf{ED}(x) \wedge \mathsf{CN}(y) \wedge \mathsf{T}(t)$ (endurant x is classified by y at time t [4])
(16) $ACF(x, y, t) \rightarrow CF(x, y, t)$ (17) $PCF(x, y, t) \rightarrow CF(x, y, t)$
(18) $ACF(x, y, t) \rightarrow \mathsf{PAO}(x) \wedge \mathsf{AT}(y)$ (ACF holds between a physical artefactual object[4] and an artefact type)

Relationship Classification Moreover, we adopt the relationship classification relation from the second approach of [5], i.e. for a relationship with arity 2 there is a relation $CF(x, y, r, t)$ (extendable to different arities) meaning 'x and y are in the relationship r during t'. Since the formalisation of relationships is not the focus of this paper, we provide only a minimal definition. However, additional axioms could be introduced to formalise a particular theory, such as those of [5,8]. Since relationships are concepts, relationship classification implies the existence of an instance (RELI) that is classified by the relationship concept. The relation

---

the definition in [4] that a concept be defined by a single description nor does it impact the ability for different information objects to express a formal description in different ways.

[4] For simplicity we consider only physical artefacts in this paper; however, the formalisation presented can be easily extended to consider abstract artefacts such as Information Artefacts.

instance is *specifically constantly dependent* (SD) [3] on the objects in the relations (i.e. the relation can only be present while the related objects are present).

(19) $CF(x, y, r, t) \rightarrow \mathsf{ED}(x) \wedge \mathsf{ED}(y) \wedge \mathsf{REL}(r) \wedge \mathsf{T}(t)$

(20) $CF(x, y, r, t) \rightarrow \exists i\, \mathsf{RELI}(i) \wedge PCF(i, r, t) \wedge SD(i, x) \wedge SD(i, y)$

Constraint Evaluation Constraints and descriptions can be applied (i.e. evaluated for truth) in the context of an object. This is denoted by $\phi(x, t)$, which means 'the constraint/description ϕ is satisfied by x during t.' Therefore, the requirement that the constraints of concepts are satisfied by the objects classified by them can be considered as follows:

(21) $ACF(x, y, t) \wedge DF(y, \phi) \rightarrow \bigwedge_{\psi, NEC(\psi,\phi)} \psi(x, t)$

(All *necessary* constraints must be satisfied by an object that is artefactually classified by a concept)

(22)* $PCF(x, y, t) \wedge DF(y, \phi) \rightarrow \bigwedge_{\psi, P(\psi,\phi)} \psi(x, t)$

(*All* constraints must be satisfied by an object that is proper classified by a concept)

The problem with using (22) for proper classification is that it only works with strict classification, i.e., when each level of classification is fully described by the immediate level above. However, when working with multiple levels of classification, constraints may be defined that relate qualities, etc. across multiple levels and that cannot be satisfied at an intermediate level. This occurs with technical artefacts where the designs, e.g., specify particular requirements or nominal qualities, such as 'operating temperature', that the physical artefact is to fulfil. In traditional representations, the operating temperature would be defined as a necessary quality for the artefact kind 'Pump', while a subclass representing an artefact species of pump (i.e., a pump model) would constrain the range of the quality to enforce the requirement. This implies that the physical pump would cease to be a pump if its physical temperature strayed outside of the range of its design, or various other nonsensical outcomes depending on formalisation and commitments in use. This is due to conflating the qualities of the design (the requirements), with the qualities of the physical artefact.

A multi-level classification system separates these: there are distinct concepts for the artefact kind 'Pump', the artefact species (or pump models), and the concept of 'Pump Model'. The latter describes what all designs of pump—i.e., the artefact species which subclass 'Pump'—are to specify, such as the 'operating temperature' requirement distinct from the 'temperature' quality of all pumps. The two are naturally linked via a constraint, such as 'the temperature of a pump *should* remain within the operating temperature range defined by its pump model.'

To do so in a general fashion, the constraint between the requirement at one level and the quality at another must be definable. Such a constraint is not satisfied at the concept of 'Pump Model', nor at a specific model of pump, but rather at the level of the physical pump artefact. Therefore, proper classification cannot simply demand that *all* of the described constraints be satisfied by the classified object as it would prevent the definition of such multi-level constraints.

Artefactual classification does not exhibit this issue as it always classifies an artefact, never a concept, and thus always requires a complete description.

To support constraints across multi-level classification hierarchies, we replace (22) with (23) to take into account the nature of the constraints and whether or not they can be evaluated in an object's context. Doing so maintains consistency when propagating constraints across classification levels via domain specific relationships by only evaluating constraints if the classified object is not a concept (hence fully described) or if the (abstract) quality or relationship has been previously introduced. The idea stems from the notion that the constraint that introduces the use of a quality or relationship is not the concept on which it should be applied, rather it should be applied at a lower level of abstraction. When applying the basic framework, where each level is fully described, the evaluation of all constraints will occur at the immediate level below as expected, as all the qualities and relationships will be introduced by the classifying concept.

(23) $PCF(x,y,t) \wedge DF(y,\phi) \rightarrow \bigwedge_{\psi,EV(\psi,\phi,x)} \psi(x,t)$

(24) $EV(\psi,\phi,x) \leftrightarrow P(\psi,\phi) \wedge \big(\neg\mathsf{CN}(x) \vee$ (Constraint ψ of description ϕ can be evaluated in the context of object x)
$(\mathsf{CN}(x) \wedge \forall q\,[US(q,\psi) \wedge q \in \mathcal{Q}_A \rightarrow \neg INTR(q,\phi)]) \vee$
$(\mathsf{CN}(x) \wedge \forall r\,[US(r,\psi) \wedge \mathsf{REL}(r) \rightarrow \neg INTR(r,\phi)])\big)$

Where *INTR* means 'introduced by' and *US* means 'used by', which are formalised later for the internal structure of descriptions and constraints.

Subconcepts A concept can be a subconcept of another, similar to subsumption between unary predicates [4]. Everything that is classified by a subconcept is classified by the parent concept. In particular, if an object is artefactually classified or proper classified by a concept, then it must also be artefactually classified or proper classified by any parent concepts. As a result, an object must satisfy the descriptions of all parent concepts (28, 29).

(25) $SB(x,y) \rightarrow \mathsf{CN}(x) \wedge \mathsf{CN}(y)$ (x is a subconcept of the concept y)
(**d**2) $SB(x,y) \triangleq \forall t\big[\exists z[CF(z,x,t)] \wedge \forall z[CF(z,x,t) \rightarrow CF(z,y,t)]\big]$ (adapted from [4])
(26) $SB(x,y) \wedge SB(y,z) \rightarrow SB(x,z)$ (Subconcepts are transitive)
(27) $\mathcal{P}(x,y,t) \wedge SB(y,z) \rightarrow \mathcal{P}(x,z,t)$, for $\mathcal{P} \in \{ACF, PCF\}$
(28) $ACF(x,y,t) \wedge SB(y,z) \wedge DF(z,\phi) \rightarrow \bigwedge_{\psi,NEC(\psi,\phi)} \psi(x,t)$ (from 21, 27)
(29) $PCF(x,y,t) \wedge SB(y,z) \wedge DF(z,\phi) \rightarrow \bigwedge_{\psi,EV(\psi,\phi)} \psi(x,t)$ (from 23, 27)

Constraint Structure Constraints must be defined in some constraint language, which we leave flexible in this framework. However, there are minimum requirements. Since the descriptions (and, hence, constraints) must be anchored in the ontology [4], the language must support the ability to: (a) reference the primitives of the ontology; (b) reference well known concept names and object names from the domain of quantification (treated as constants); (c) reference a special constant, e.g. SELF, representing the contextual object with respect to which the constraints are evaluated; (d) refer to qualities and values (regions of a quality space); (e) include logical operators and quantifiers, such as conjunction, disjunction, universal quantification, existential quantification, etc.

These requirements allow defining key concepts and constraints: (a) relationships linked to primitive predicates, e.g., domain specific part-of relationship associated with the primitive parthood predicate (P); (b) constraints on the existence of a quality on the classified objects, e.g., wheels require a diameter; (c) constraints on the values of a quality, e.g., 10 inch wheels have a diameter within the 10 inch region of the quality space. This may be inexact to allow for tolerances, e.g., $10 \pm {}^{1}\!/_{4}$ inches (i.e., nominal qualities); (d) constraints on the existence of entities of certain types, e.g., a concept may require a specific relationship to an object of another type, while relationships may have constraints on the existence of mediating entities; (e) constraints based on temporal aspects and perdurants such as process instances, e.g., some constraints on a piece of equipment may only apply while the equipment is participating in a "running" process.

To keep the theory agnostic of any particular constraint language, we re-purpose the 'used by' relation (US) of [4] to characterise the contents of constraints in terms of the qualities, relationships, and concepts that they reference. The re-purposed relation is formalised almost identically to the original, allowing the original to be derived from the contents of the description, rather than as primitive on descriptions themselves. In particular, the axiom that enforces the use of a concept in its own description (33) remains applicable as it embodies the idea that the constraints are in the context of an object of that concept. Moreover, each constraint uses the defined concept (34) since the reference to the special constant SELF implicitly uses the concept of the description and is the starting point of each constraint. In addition, we specify the 'introduced by' relation ($INTR$) to identify constraints that introduce qualities or relations, not just use them. This allows for the identification of the point in a classification hierarchy that a quality or relationship is introduced and not just referenced. When dealing with a specific constraint language, this relation should be defined in terms of that language.

(30) $US(x,y) \rightarrow (x \in \mathcal{Q} \vee \mathsf{CN}(x)) \wedge (\mathsf{DS}(y) \vee \mathsf{CSTR}(y))$ (The quality or concept[5] x is used by the description or constraint y)

(31) $INTR(x,y) \rightarrow (x \in \mathcal{Q} \vee \mathsf{REL}(x)) \wedge (\mathsf{FD}(y) \vee \mathsf{CSTR}(y))$ (The quality or relationship x is introduced by y)

(32) $INTR(x,y) \rightarrow US(x,y)$ (An introduced quality/relationship is used by the constraint)

(33) $DF(x,y) \rightarrow US(x,y)$ (A concept is used by the description that defines it [4])

(34) $DF(x,y) \wedge P(c,y) \rightarrow US(x,c)$ (All parts/constraints of a description use the concept the description defines)

(35) $US(x,y) \rightarrow (PRE(y,t) \rightarrow PRE(x,t))$ (Used concepts must be present when the using entity is present [4])

(36) $US(x,y) \wedge \mathsf{FD}(y) \leftrightarrow \exists c\, \mathsf{CSTR}(c) \wedge P(c,y) \wedge \mathsf{FD}(y) \wedge US(x,c)$

(37) $INTR(x,y) \wedge \mathsf{FD}(y) \leftrightarrow \exists c\, \mathsf{CSTR}(c) \wedge P(c,y) \wedge INTR(x,c)$

A special handling of quality types is required as they are universals. Here we use a treatment similar to DOLCE [3] which considers a second order axiom as syntactic sugar for a finite list of first-order axioms. This can be achieved since, for a given ontology, we can assume a finite set of quality types, $\mathcal{Q}$. In addition, we can posit sets representing the partitioning of quality types as physical, abstract, or temporal as $\mathcal{Q}_P$, $\mathcal{Q}_A$, and $\mathcal{Q}_T$, respectively. However, this approach mixes constraints over universals with elements in the domain of quantification and requires modification of the universals of the ontology to incorporate domain specific quality types. An alternative would integrate quality types into the domain

---

[5] Since relationships are concepts they do not have to be explicitly mentioned.

of quantification in a similar fashion to concepts and relationships; this will be performed in future work.

Example Constraints Consider the simplified representation of a type of pump (such as a particular model of centrifugal pump) and its specification. It may include constraints on necessary parts[6], for example, an impeller of a certain type. The impeller itself may define necessary constraints such as its diameter being of a certain dimension (within some tolerance). Moreover, the pump may include constraints on its operating temperature and flow rate[7] such that they must stay within certain ranges but can be violated. For example, if the connecting pipe is broken the flow rate will drop without affecting the object's status as a pump. The referenced quality types (e.g., $\mathsf{operating\text{-}temp}$) are assumed to have been incorporated into the ontology. Example 1 illustrates these constraints using a first-order logic-like language where terms in SMALL CAPS represent names of concepts and qualities that are referenced in the constraint. As the constraints are reified in the domain of quantification, they are illustrated as symbols with their content defined in the boxes. In addition, the following shorthand is used for classification: $\textsc{cn}(x,t) \rightarrow \mathit{CF}(x, \textsc{cn}, t)$, $\textsc{r}(x,y,t) \rightarrow \mathit{CF}(x,y,\textsc{r},t)$ The relations qt and ql from DOLCE [3] indicate that something is a quality and a quale (i.e., a value of a quality), respectively. For brevity, the quantification of time inside constraint expressions is in the context of the object for which the constraint is evaluated. When no explicit quantification is specified it is implicitly: $\forall t\, \mathit{PRE}(\textsc{self}, t)$. For example, in $\psi_{\textsc{pump1}}$, a pump must have an impeller only for those times at which the pump is present.

Example 1 (Definition of a pump concept and associated constraints)

$$\mathsf{AS}(\textsc{pump}) \wedge \mathit{DF}(\textsc{pump}, \phi_{\textsc{pump}}) \wedge \mathit{NEC}(\psi_{\textsc{pump1}}, \phi_{\textsc{pump}}) \wedge \mathit{OBG}(\psi_{\textsc{pump2}}, \phi_{\textsc{pump}})$$

$$\wedge\, \mathit{OBG}(\psi_{\textsc{pump3}}, \phi_{\textsc{pump}})$$

$$\psi_{\textsc{pump1}} = \boxed{\exists x\, \textsc{has-part}(\textsc{self}, x, t) \wedge \textsc{impeller}(x, t)}$$

$$\psi_{\textsc{pump2}} = \boxed{\begin{array}{l} \exists q\, qt(q, \textsc{self}) \wedge \mathsf{operating\text{-}temp}(q) \\ \quad \wedge\, ql(v, q, t) \wedge P(v, \underline{\mathsf{100^\circ\text{-}200^\circ C}}, t) \end{array}}$$

$$\psi_{\textsc{pump3}} = \boxed{\exists q\, qt(q, \textsc{self}) \wedge \mathsf{flow\text{-}rate}(q) \wedge ql(v, q, t) \wedge P(v, \underline{\mathsf{10 \pm 0.5\ L/m}}, t)}$$

$$\mathsf{AS}(\textsc{impeller}) \wedge \mathit{DF}(\textsc{impeller}, \phi_{\textsc{impeller}}) \wedge \mathit{NEC}(\psi_{\textsc{impeller1}}, \phi_{\textsc{impeller}})$$

$$\psi_{\textsc{impeller1}} = \boxed{\exists q\, qt(q, \textsc{self}) \wedge \mathsf{diameter}(q) \wedge ql(v, q, t) \wedge P(v, \underline{\mathsf{10 \pm {}^{1}\!/\!_{4}''}}, t)}$$

$$\mathsf{REL}(\textsc{has-part}) \wedge \mathit{DF}(\textsc{has-part}, \phi_{\textsc{has-part}}) \wedge \mathit{NEC}(\psi_{\textsc{has-part1}}, \phi_{\textsc{has-part}})$$

$$\psi_{\textsc{has-part1}} = \boxed{\textsc{self}(x, y, t) \rightarrow P(y, x, t)}$$

We now briefly introduce the SLICER framework before defining how it can be integrated to improve the granularity of the descriptions.

---

[6] For simplicity of the example we ignore that parts of an artefact can be replaced.

[7] Such constraints would only apply while the pump is running; however, since we do not cover (artefactual) processes in the present work for brevity, such distinctions are not made.

3. Using the SLICER Relationship Framework for Relationship Concepts

The SLICER (Specification with Levels based on Instantiation, Categorisation, Extension, and Refinement) framework was developed in the context of complex domain models of the engineering life cycle to reduce the modelling load via flexible meta-level modelling techniques.

A hierarchy of layers is used to separate ontological from representational ("linguistic") aspects, a concept known as multilevel modelling. In the engineering life cycle, a higher level generally expresses the relationship between an entity and its definition (or description), correlating with the intentional design process [9].

A *level of description* can be established either by instantiation, or by enriching the vocabulary used to formulate the descriptions. This corresponds to the concept of *extension* in specialisation hierarchies [10]: if a subclass describes additional properties (attributes, relationships, etc.) then these properties can be used to impose constraints on its specification and behaviour.

To support the purpose of describing joint metamodels in interoperability scenarios, SLICER is based on a flexible notion of levels identified by applying the semantic relationships below.

Instantiation and Specialisation Levels of description are dynamically derived based on finer distinctions of the instantiation and specialisation relations. Specialisation relationships *extend* the original class (by adding attributes, associations, or behaviour, i.e., constraints or state change differentiation, $SpecX$) or *refines* it (by adding granularity to the description, $SpecR$). Of the two, only $SpecX$ introduces a new model level.

Similarly, instantiation is characterised as *Instantiation with Extension* ($InstX$, allowing additional properties, etc.) or *Normal Instantiation* ($InstN$). Instantiation always introduces additional model levels. Objects created through $InstN$ cannot be instantiated further and form the most basic-level of individuals in a model.

Categories are concepts providing external ("secondary") grouping of entities based on common properties and/or explicit enumeration of members. We do not discuss them further here.

Specifications are expressed via the *Subset by Specification* (SbS) relation. The specification class (for example EquipmentModel) exists at the same level as the type it refers to as it can define constraints with respect to that type. Specifications and Categories relate to two common ways in which powertypes are applied [2].

Descriptions A description, e.g. a set of constraints, can refer only to the properties specific to its object (or, if the description is for a specification, the properties of the type associated with the specification) and are inherited through specialisation, while instances of a type must satisfy its description.

This framework can now be used on top of the basic formalisation from Section 2. That formalisation allowed for the definition of concepts, their (intended) instances, and the checking of constraints between an entity and its classifying concepts. As a result, the definitions of SLICER relationships can be reformulated within the ontological framework, by defining the relationships as instances of REL, linking them to appropriate primitive relations, and incorporating the SLICER axioms as constraints within their descriptions. This way different information representations, including information models and data models at different levels of expressivity, can be formalised within the same ontological framework.

The following definitions validate the ontological framework against the requirements of specifying SLICER relationships.

3.1. General Instantiation and Specialisation Relationships of SLICER

The formulation of the core relationships of general instantiation (*Inst*) and specialisation (*Spec*) relationships is shown in Eqs. (38) and (39), respectively. Below, *Inst* is a relationship concept (REL) defined by (DF) the description $\phi_{\textsc{Inst}}$ with the necessary constraint (NEC) labelled $\psi_{\textsc{Inst}1}$. $\psi_{\textsc{Inst}1}$ is a reified expression indicating that instances of the relationship imply adherence to the standard classification relation, cf. (15), tying the domain relationship to the ontological primitives. The general specialisation relationship is defined analogously.

$$\begin{gathered}\mathsf{REL}(\textsc{Inst}) \land DF(\textsc{Inst}, \phi_{\textsc{Inst}}) \land NEC(\psi_{\textsc{Inst}1}, \phi_{\textsc{Inst}}),\\ \psi_{\textsc{Inst}1} = \boxed{\textsc{self}(x,y,t) \to CF(x,y,t)}\end{gathered} \tag{38}$$

$$\begin{gathered}\mathsf{REL}(\textsc{Spec}) \land DF(\textsc{Spec}, \phi_{\textsc{Spec}}) \land NEC(\psi_{\textsc{Spec}1}, \phi_{\textsc{Spec}}),\\ \psi_{\textsc{Spec}1} = \boxed{\textsc{self}(x,y,t) \to SB(x,y,t)}\end{gathered} \tag{39}$$

3.2. Relationships Incorporating Extension and Refinement

SLICER introduces more specific relationships based on whether an object *extends* (adding additional attributes, behaviour, etc.), *refines* (adding granularity, e.g. by restricting the range of an attribute), and/or *instantiates* (i.e. assigns values to its attributes) another [2]. These relationships can be defined as shown in Eqs. (40) and (41). All specialisations may include refinement, while *SpecX* must incorporate additional attributes, i.e. qualities or relationships in this context (refer Eq. (41)). To support this we add a constraint that enforces the propagation of constraints across all *Spec* relationships such that necessary constraints remain necessary and obligations remain obligations (Eq. (40)). This remains consistent with the evaluation of descriptions of parent types during classification. Although it would be symmetrical with *SpecX* to enforce refinement in *SpecR*, doing so would disallow the ability to extend the vocabulary of concepts in the case where no refinement to the modelled attributes is included. For the same reason, we have not defined the identity of concepts based on their intension, that is, the constraints included in their descriptions. (However, a specific application could choose to include additional axioms to enforce such a constraint.)

$$\begin{gathered}DF(\textsc{Spec}, \phi_{\textsc{Spec}}) \land NEC(\psi_{\textsc{Spec}2}, \phi_{\textsc{Spec}}),\\ \psi_{\textsc{Spec}2} = \boxed{\begin{array}{c}\textsc{self}(x,y,t) \land DF(x,\phi_x) \land DF(y,\phi_y) \to\\ \forall c\,[\mathcal{P}(c,\phi_y) \to \mathcal{P}(c,\phi_x)]\end{array}} \text{ for } \mathcal{P} \in \{NEC, OBG\}\end{gathered} \tag{40}$$

$$\begin{aligned}
&\mathsf{REL}(\textsc{SpecX}) \wedge \mathit{DF}(\textsc{SpecX}, \phi_{\textsc{SpecX}}) \wedge \mathit{NEC}(\psi_{\textsc{SpecX}1}, \phi_{\textsc{SpecX}}) \\
&\qquad \wedge \mathit{NEC}(\psi_{\textsc{SpecX}2}, \phi_{\textsc{SpecX}}), \\
&\psi_{\textsc{SpecX}1} = \boxed{\textsc{self}(x,y,t) \rightarrow \textsc{Spec}(x,y,t)}, \\
&\psi_{\textsc{SpecX}2} = \boxed{\begin{array}{r}\textsc{self}(x,y,t) \wedge \mathit{DF}(x,\phi_x) \wedge \mathit{DF}(y,\phi_y) \rightarrow \\ \exists z\,(\mathsf{REL}(z) \vee z \in \mathcal{Q}) \wedge \mathit{US}(z,\phi_x) \wedge \neg\, \mathit{US}(z,\phi_y)\end{array}}
\end{aligned} \tag{41}$$

A similar characterisation can be given of *InstX* and *InstN*, as illustrated in Eqs. (42) and (43). The intended application of SLICER is for the definition of artefacts, so we can define *InstN* in terms of artefactual classification.[8]

$$\begin{aligned}
&\mathsf{REL}(\textsc{InstX}) \wedge \mathit{DF}(\textsc{InstX}, \phi_{\textsc{InstX}}) \wedge \mathit{NEC}(\psi_{\textsc{InstX}1}, \phi_{\textsc{InstX}}) \\
&\qquad \wedge \mathit{NEC}(\psi_{\textsc{InstX}2}, \phi_{\textsc{InstX}}), \\
&\psi_{\textsc{InstX}1} = \boxed{\textsc{self}(x,y,t) \rightarrow \textsc{Inst}(x,y,t) \wedge \mathit{PCF}(x,y,t)}, \\
&\psi_{\textsc{InstX}2} = \boxed{\begin{array}{r}\textsc{self}(x,y,t) \wedge \mathit{DF}(x,\phi_x) \wedge \mathit{DF}(y,\phi_y) \rightarrow \\ \exists z\,(\mathsf{REL}(z) \vee z \in \mathcal{Q}) \wedge \mathit{US}(z,\phi_x) \wedge \neg\, \mathit{US}(z,\phi_y)\end{array}}
\end{aligned} \tag{42}$$

$$\begin{aligned}
&\mathsf{REL}(\textsc{InstN}) \wedge \mathit{DF}(\textsc{InstN}, \phi_{\textsc{InstN}}) \wedge \mathit{NEC}(\psi_{\textsc{InstN}1}, \phi_{\textsc{InstN}}) \\
&\psi_{\textsc{InstN}1} = \boxed{\textsc{self}(x,y,t) \rightarrow \textsc{Inst}(x,y,t) \wedge \mathit{ACF}(x,y,t)}
\end{aligned} \tag{43}$$

This characterisation implies that only concepts can be in an *InstX* relationship with another concept, while non-concepts are the instances in *InstN* relationships. Since non-concepts do not have descriptions, they cannot be extended with constraints on additional quality types or relationships. Since the ontological framework ensures that qualities have values (or quales), the assignment of values to qualities does not need to be included in the constraints of *InstN*. Similarly, given the axioms adopted for relations above, the existence of a value for a relationship required by a constraint is already enforced; therefore, it is not necessary to include a constraint for this in the description of *InstN*.

3.3. Subset by Specification

Another important relation in SLICER is Subset by Specification (SbS), which is a form of powertyping relation that states the instances of the specification type are subconcepts of the associated concept [2]. Such a construct is frequently encountered in design and manufacturing settings. A minimal form of this relationship can be defined within the ontological framework as shown in Eq. (44).

---

[8] This may not be the case for the full framework including roles and functions.

$$\begin{aligned}
&\mathsf{REL}(\textsc{SbS}) \land \mathit{DF}(\textsc{SbS}, \phi_{\textsc{SbS}}) \land \mathit{NEC}(\psi_{\textsc{SbS}1}, \phi_{\textsc{SbS}}) \land \mathit{NEC}(\psi_{\textsc{SbS}2}, \phi_{\textsc{SbS}}),\\
&\psi_{\textsc{SbS}1} = \boxed{\textsc{self}(x, y, t) \to \mathsf{CN}(x) \land \mathsf{CN}(y)}, \qquad (44)\\
&\psi_{\textsc{SbS}2} = \boxed{\textsc{self}(c, c', t) \land \mathit{Inst}(x, c, t) \to \textsc{Spec}(x, c', t)}
\end{aligned}$$

3.4. Constraint Propagation Across Instantiation Relationships

Finally, SLICER introduces an intuitive method of propagating constraints across multiple levels of instantiation/classification to where they can be evaluated. The propagation is determined by whether a constraint can be evaluated for the current object based on the presence or absence of appropriate attributes [2]. Within the ontological framework we can make a similar distinction based on the types of qualities and relations used by a constraint along with the *INTR* relation.

For example, a higher level concept may define constraints based on physical qualities; however, concepts are non-physical and cannot have physical qualities in DOLCE [3]. Therefore, when a concept instantiates the higher level concept, the constraints using physical qualities are propagated to the instantiating concept. Then, when a physical object instantiates the bottom-most concept, the constraints can be evaluated in the context of a physical object. A similar situation occurs for abstract and temporal qualities, except that other non-physical objects (not just concepts) can have abstract qualities. Therefore, the propagation only occurs for the abstract qualities that have not yet been introduced. Eq. (45) illustrates how the propagation can be defined on the descriptions of the instantiation relationships using a shorthand to abstract over necessary and obligatory constraints.

$$\begin{aligned}
&\mathit{DF}(\textsc{Inst}, \phi_{\textsc{Inst}}) \land \mathit{NEC}(\psi_{\textsc{Inst}2}, \phi_{\textsc{Inst}}),\\
&\psi_{\textsc{Inst}2} = \boxed{\begin{aligned}
&\textsc{self}(x, y, t) \land \mathit{DF}(x, \phi_x) \land \mathit{DF}(y, \phi_y) \to\\
&\quad \forall c, q\, [\mathcal{P}(c, \phi_y) \land \mathit{US}(q, c) \land q \in \mathcal{Q}_P \cup \mathcal{Q}_T \to \mathcal{P}(c, \phi_x)] \land\\
&\quad \forall c, q\, [\mathcal{P}(c, \phi_y) \land \mathit{US}(q, c) \land \neg \mathit{INTR}(q, \phi_y) \land q \in \mathcal{Q}_A \to \mathcal{P}(c, \phi_x)] \land\\
&\quad \forall c, r\, [\mathcal{P}(c, \phi_y) \land \mathit{US}(r, c) \land \neg \mathit{INTR}(r, \phi_y) \land \mathsf{REL}(r) \to \mathcal{P}(c, \phi_x)]
\end{aligned}}\\
&\text{for } \mathcal{P} \in \{\mathit{NEC}, \mathit{OBG}\} \qquad (45)
\end{aligned}$$

Constraint Propagation Example Consider extending the constraints of Example 1 to utilise the SLICER model. Here the operating temperature constraint is not defined solely as operating temperature; instead, it is defined as a comparison between the values of two qualities that are introduced in different concepts such that the constraint is propagated over two instantiation relationships. The constraint is defined between the operating temperature (physical) quality, defined on PumpModel, and the (actual) temperature quality, defined on Pump (the superclass of all pump types). The operating temperature is a *design* (or *nominal*) quality that can be constrained to a specific region by the different pump models/ types. The concept definitions and their constraints are illustrated in Example 2.

Example 2 (Definition of pump concept and constraints using SLICER)

$$\begin{aligned}
&\mathsf{AK}(\textsc{Pump}) \land \mathit{DF}(\textsc{Pump}, \phi_{\textsc{Pump}}) \land \mathit{NEC}(\psi_{\textsc{Pump}1}, \phi_{\textsc{Pump}})\\
&\psi_{\textsc{Pump}1} = \boxed{\exists x\; qt(x, \textsc{self}) \land \mathsf{temperature}(x)}
\end{aligned}$$

$$\begin{aligned}
&\mathsf{CN}(\textsc{PumpModel}) \wedge \textsc{SbS}(\textsc{PumpModel}, \textsc{Pump}, t) \wedge DF(\textsc{PumpModel}, \phi_{\textsc{PumpModel}})\\
&\qquad \wedge NEC(\psi_{\textsc{PumpModel1}}, \phi_{\textsc{PumpModel}})\\
&\qquad \wedge OBG(\psi_{\textsc{PumpModel2}}, \phi_{\textsc{PumpModel}})
\end{aligned}$$

$$\psi_{\textsc{PumpModel1}} = \boxed{\exists q\ qt(q, \textsc{self}) \wedge \mathsf{operating\text{-}temp}(q)}$$

$$\psi_{\textsc{PumpModel2}} = \boxed{\begin{array}{l} qt(q_{ot}, \textsc{self}) \wedge \mathsf{operating\text{-}temp}(q_{ot}) \wedge ql(v_{ot}, q_{ot}, t) \wedge \\ \quad qt(q_t, \textsc{self}) \wedge \mathsf{temperature}(q_t) \wedge ql(v_t, q_t, t) \rightarrow P(v_t, v_{ot}, t) \end{array}}$$

$$\begin{aligned}
&\mathsf{AS}(\textsc{C12Pump}) \wedge \mathit{SpecX}(\textsc{C12Pump}, \textsc{Pump}, t) \wedge \mathit{InstX}(\textsc{C12Pump}, \textsc{PumpModel}, t)\\
&\qquad \wedge DF(\textsc{C12Pump}, \phi_{\textsc{C12Pump}}) \wedge NEC(\psi_{\textsc{C12Pump1}}, \phi_{\textsc{C12Pump}})\\
&\qquad \wedge NEC(\psi_{\textsc{C12Pump2}}, \phi_{\textsc{C12Pump}})
\end{aligned}$$

$$\psi_{\textsc{C12Pump1}} = \boxed{\exists x\ \textsc{has-part}(\textsc{self}, x, t) \wedge \textsc{impeller}(x, t)}$$

$$\psi_{\textsc{C12Pump2}} = \boxed{qt(q, \textsc{self}) \wedge \mathsf{operating\text{-}temp}(q) \rightarrow ql(\underline{100^\circ\text{-}200^\circ\mathrm{C}}, q, t)}$$

Therefore, to be a proper instance of C12Pump, a physical pump's temperature must be consistent with the operating temperature range of 100°-200°C. If the obligation is not fulfilled, the pump is still considered to be a C12Pump due to the artefactual classification; however, it does not conform to its specification.

The constraints defined on PumpModel will be propagated to the description of C12Pump via the instantiation relationship, thereby making it an obligatory constraint for all pumps. This demonstrates the ability to tailor the semantics in the ontology to specific domains entirely within the domain of discourse. Different domains may use different sets of semantics side-by-side: indeed, as defined here, the SLICER semantics are specifically suited to artefacts.

4. Related Work

Wang et al. [11] have conducted ontological analysis of software systems based on a requirements engineering perspective. They defined multiple levels of detail, bottoming out in code, where each level is *constitutedBy* lower-level pieces of software in a relationship defined by Baker [12]. Moreover, each level (excluding code) is associated with a specification or description. To do so they define a relationship *intendedToImplement* that corresponds roughly to our ACF relation; the "intended" implying that the implementation may not be exactly correct. Relations *intendedToSatisfy* and *presupposes* are used to relate different levels of requirements and separate out environment assumptions. However, the specifications are not broken down further and no axiomatisation is given.

Guarino and Melone [13] informally discuss the ontological status of design objects, based on the viewpoint of architects, and therefore assuming that design objects represent physical artifacts. The basic role is that of *design element* which, installed in a particular position, serves as a *design component*. A *conventional system component*, as in [6] is a particular location considered by the designer and a *physical system component* is an actual physical object resulting from the design. While Guarino and Melone identify these different design objects, they do not consider the definition or content of the specifications.

The work by Sanfilippo et al. [14] is most similar to our own, focusing on a structural decomposition of design objects that bottoms out in quality spaces. The axiomatisation of (artefactual) concepts is based on some of the same basis as our work such as DOLCE [3] and previous work on roles [4]. The characterisation differentiates and compares design concepts vs. requirements concepts, but does not consider the relationship to instances of the concepts. In contrast, we differentiate necessary and obligatory constraints in the context of how they are fulfilled by artefact instances. In addition, our characterisation of concepts goes further than a simple group of properties as we explicitly associate the concepts with constraints.

More recently, Sanfilippo et al. [9] investigated the foundational ontological basis of nominal and actual qualities through three possible representations: Nominal qualities as qualities, Nominal qualities as properties, and Nominal qualities as descriptions. Nominal qualities are highly related to engineering design as they define constraints to which an artefact, with actual qualities, is expected to conform to some degree. Our approach fits into the latter representation, in which nominal qualities are defined by descriptions, where we characterise the content of the descriptions as formal, executable constraints. While we have only illustrated simple constraints, including what would be considered nominal qualities, our approach is capable of (and intended for) much more complex constraints.

The work of [15,16] does not formalise the description of an artefact, but merely its interface for assembly of cosimulation processes, but could serve as another use case addressing one specific phase of the artefact life cycle. While their GOPPRR approach supports the incorporation of different domains or information representations into a single model, there is limited to no ontological commitment.

In [17], a group of modular ontologies built on the Basic Formal Ontology upper ontology are described that cover the engineering life cycle, including ‘design specifications’, etc. The approach treats design specifications as Information Content Entities (in the terms of the BFO-conforming Information Artifact Ontology), rather than characterising their content as generic constraints that can be applied to their instances. Also, as pointed out in [9], the BFO-based approach may have difficulty representing nominal qualities (and, hence, more complex constraints) as the Information Content Entities must be defined for things that may not yet exist, which is in conflict with their definition.

5. Conclusion

In this work, we have brought together two threads of our work on large-scale model-driven interoperability: the framework for artefacts and roles, based on an extension of the DOLCE ontology [1], and the SLICER (Specification with Levels based on Instantiation, Categorisation, Extension, and Refinement) approach providing the semantic building blocks for the cleanly structured representation of industrial artefact domains in detail [2]. A key innovation of SLICER is the explicit handling of artefact *descriptions*, and based on Masolo and others’ work on social roles [4] we have provided a framework for the formalisation of descriptions that enables the modelling and management of domain specific constraints on technical artefacts. We have shown how to represent necessary and obligatory constraints in the context of evolving information, and obtained a framework capable of validating information models across multiple life cycle stages w.r.t. specific domain requirements. The framework is being incorporated in our F-Logic based transformation and validation environment [2]. Future work will use this to perform tasks such as requirements verification, incorporate the handling of functions and roles, and study the semantic SLICER relationship types as meta-ontological properties in the plant life cycle domain. Also, adopting a reduced ontological commitment following the (Constructive) Descriptions and Situations framework [18] would allow this work to be applied across different foundational ontologies in support of semantic interoperability for complex constraints.

References

[1] Jordan A, Selway M, Mayer W, Grossmann G, Stumptner M. An Ontological Core for Conformance Checking in the Engineering Life-cycle. In: Proc. FOIS 2014; 2014. .

[2] Selway M, Stumptner M, Mayer W, Jordan A, Grossmann G, Schrefl M. A conceptual framework for large-scale ecosystem interoperability and industrial product lifecycles. Data Knowl Eng. 2017;109:85-111.

[3] Masolo C, Borgo S, Gangemi A, Guarino N, Oltramari A, Schneider L. WonderWeb Deliverable D17: The WonderWeb Library of Foundational Ontologies. In: TR-NRC, Institute of Cognitive Sciences and Technology. Italy; 2003. .

[4] Masolo C, Vieu L, Bottazzi E, Catenacci C, Ferrario R, Gangemi A, et al. Social Roles and their Descriptions. In: Proc. KR '04; 2004. p. 267-77.

[5] Masolo C, Guizzardi G, Vieu L, Bottazzi E, Ferrario R. Relational roles and qua-individuals. In: Proc. AAAI Fall Symposium on Roles; 2005. p. 103-12.

[6] Guarino N. Artefactual Systems, Missing Components and Replaceability. In: Franssen M, et al., editors. Artefact Kinds. Springer; 2014. p. 191-206.

[7] Mizoguchi R, Kitamura Y, Borgo S. Towards a Unified Definition of Function. In: Proc. FOIS; 2012. p. 103-16.

[8] Guarino N, Guizzardi G. "We Need to Discuss the Relationship": Revisiting Relationships as Modeling Constructs. In: Proc. CAiSE; 2015. p. 279-94.

[9] Sanfilippo EM, et al. A Foundational View on Nominal and Actual Qualities in Engineering. In: Proc. FOIS. Cape Town; 2018. .

[10] Schrefl M, Stumptner M. Behavior Consistent Specialization of Object Life Cycles. ACM TOSEM. 2002;11(1):92-148.

[11] Wang X, Guarino N, Guizzardi G, Mylopoulos J. Towards an Ontology of Software. In: Proc. FOIS 2014. Rio de Janeiro; 2014. .

[12] Baker LR. The Ontology of Artifacts. Philos Explor. 2004;7:99-112.

[13] Guarino N, Melone MRS. On the ontological status of design objects. In: 1st Workshop on AI and Design at AIIA; 2015. .

[14] Sanfilippo E, Masolo C, Porello D. Design knowledge representation: an ontological perspective. In: 1st Workshop on AI and Design at AIIA; 2015. .

[15] Wang H, Wang G, Lu J, Ma C. Ontology Supporting Model-Based Systems Engineering Based on a GOPPRR Approach. In: Proc. WorldCIST'19. Cham: Springer; 2019. p. 426-36.

[16] Lu J, Wang G, Törngren M. Design Ontology in a Case Study for Cosimulation in a Model-Based Systems Engineering Tool-Chain. IEEE Systems Journal. 2020;14(1):1297-308.

[17] Otte JN, et al. An ontological approach to representing the product life cycle. Applied Ontology. 2019;14(2):179-97. Available from: https://doi.org/10.3233/AO-190210.

[18] Gangemi A. Norms and plans as unification criteria for social collectives. Auton Agents Multi Agent Syst. 2008;17(1):70-112.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210369

Semantic Analysis of Winograd Schema No. 1

Brandon BENNETT [a,1]
[a] *School of Computing, University of Leeds, UK*

Abstract. The Winograd Schema Challenge is a general test for Artificial Intelligence, based on problems of pronoun reference resolution. I investigate the semantics and interpretation of Winograd Schemas, concentrating on the original and most famous example. This study suggests that a rich ontology, detailed commonsense knowledge as well as special purpose inference mechanisms are all required to resolve just this one example. The analysis supports the view that a key factor in the interpretation and disambiguation of natural language is the preference for *coherence*. This preference guides the resolution of co-reference in relation to both explicitly mentioned entities and also implicit entities that are required to form an interpretation of what is being described. I suggest that assumed identity of implicit entities arises from the expectation of coherence and provides a key mechanism that underpins natural language understanding. I also argue that conceptual ontologies can play a decisive role not only in directly determining pronoun references but also in identifying implicit entities and implied relationships that bind together components of a sentence.

Keywords. natural language semantics, pronoun resolution, coherence, ontology

1. Introduction

The Winograd Schema Challenge (WSC) was proposed by Levesque *et al.* [1] as an updated form of the Turing Test. It provides a method for evaluating AI systems by means of a text processing problem, whose solution seems to require both understanding of the meaning of natural language, background knowledge of physical and social situations and commonsense reasoning. Specifically, the WSC is the task of solving pronoun resolution problems having similar form to the following paradigm case (originally considered by Terry Winograd [2]):

WS1. ***The city councilmen*** *refused* ***the demonstrators*** *a permit because* **they** **[***feared/ advocated***]** *violence.*

Here, the pronoun to be resolved is ‘**they**’, and its possible referents are ‘***the city councilmen***’ and ‘***the demonstrators***’.

---

[1] Corresponding Author: Brandon Bennett, E-mail: B.Bennett@leeds.ac.uk. Contributions of my collaborators Suk Joon Hong and Judith Clymo and extremely useful suggestions from the paper’s referees are gratefully acknowledged.

Each schema actually corresponds to *two*[2] problem cases, which differ in the choice of one of two alternative words or phrases, indicated here by the notation '[A/B]'. So **WS1** specifies the problems of determining the referent of the pronoun '**they**' in either of the sentences **WS1a** and **WS1b** resulting from selecting each of the two alternatives. The reason why two alternatives are given, is to guard against the possibility that the anaphora resolution can be accomplished by means of some structural analysis of the sentence that can be done without any consideration of the meaning of the sentence. Since the two alternatives are syntactically identical, apart from one word (or short phrase), but imply different resolutions of the pronoun, this prevents the resolution being determined purely by the syntactic category of the co-referring expression — or so we may hope.

Although WSC was proposed to provide a general test for AI rather than stipulating how the challenge should be addressed, the paper's conclusion advocates an approach based on knowledge representation:

> *"While this approach (KR) still faces tremendous scientific hurdles, we believe it remains the most likely path to success. That is, we believe that in order to pass the WSC, a system will need to have commonsense knowledge about space, time, physical reasoning, emotions, social constructs, and a wide variety of other domains."* [3]

In the current paper I examine **WS1** from the point of view of logic and semantics, and attempt to identify structures and principles by which WS examples can be resolved. My aim is to provide illustrations and arguments supporting the following views:

- The semantic and background knowledge and types of inference required to resolve a WS can be extremely complex.
- Coherence (and cohesion) principles are key to natural language understanding.
- Natural language interpretation is heaviliy constrained and enabled by semantic type and role relationships.
- Connections between explicitly mentioned objects and concepts are mediated by the existence of *implied entities*, as well as those explicitly mentioned.
- Ontology provides a means of specifying and identifying the relevant semantic types, roles and entities required to establish coherence preference and make inferences based on these preferences.
- Despite it being clear that enormous difficulties arise when the problem of natural language understanding by means of Ontologies and KRR techniques, this is still a good approach.

1.1. Previous Work

Although the WSC was designed primarily with KR type approaches in mind, it seems that the problem has received more attention from researchers using ML techniques. Probably the first fully automated WSC resolving system was that of [4] which used a SVM algorithm working with several linguistic features, some of which are based on semantic features relationships between words within in the WS sentences. In recent work, researchers using methods based on neural language models such as BERT [5] and RoBERTa [6] have demonstrated high statistical accuracy in resolving Winograd

---

[2] It is possible to devise examples with more than two cases but for simplicity of exposition we assume that we only deal with Schemas with two cases.

schemas [7, 8, 9]. An accuracy of 90% for the original WSC problem set (WSC273) is reported in [9]. However, performance goes down significantly on the larger WSC data sets [9, 10]. But despite apparent the high accuracy of BERT-based solutions, there are several reasons to suspect that BERT's understanding of sentences is superficial. It does not work well on sentences involving function words such as negation [11], and lacks of robustness to with respect to semantically insignificant variations of input sentences (e.g. cases where just proper names are changed [7, 12]). And of course, another shortcoming of the ML approaches is that they do not give any kind of explanation of their answers.

KR methods resolve Winograd schemas by creating a logical representation of a sentence and relevant background knowledge and applying inference rules. The advantage of KR is that it can give meaningful explanations for the answers. [13] define "correlation calculus" to resolve Winograd schemas by adding a novel *correlation* connective to first-order logic. However, this method requires that WSC sentences be accurately translated into first-order formulae and all relevant background knowledge needs to be manually defined in the form of correlation calculus axioms. [14] tackles the WSC by using a semantic parser (K-Parser) to extract semantic relationships from sentences and match them to identity rules that can be automatically extracted from text corpora. However, rules that successfully resolve the pronoun are found in less than half of the cases.

2. 'Coherence' and its Application to Pronoun Resolution

Informally, we may say that a text or dialogue is *coherent* if 'it fits together well'. This phrase describes language as if it were self-assembled furniture, which may be a good metaphor. But what does 'fitting together' mean in the case of language? There is no simple answer to this question. The ways in which components of language fit together are many and varied. In this paper I explore some kinds of coherence that I believe to be particularly important in natural language understanding, but, presumably because of their somewhat covert mode of operation, have not been given the attention they deserve.

As was so convincingly argued by Grice [15], and is now generally accepted, the interpretation of language is greatly dependent upon and conditioned by various principles that arise from the cooperative nature of communication. Such principles enable language to be understood in a way that is far less ambiguous than would be the case if we relied purely on the explicitly asserted content of linguistic expressions. *Coherence* is a structural property of language rather than a maxim of communication. However, it certainly plays a role in satisfying Gricean maxims, especially those of clarity and orderliness. Several researchers have suggested that coherence is a key factor in natural language understanding and have also tried to characterise more precisely what is meant by coherence and to identify [16] or even measure [17] coherence in language samples. Coherence is often considered to arise where successive clauses or sentences refer to the same things. Hence coherence is associated with *co-reference*.[3]

Hobbs [16] applied the idea of coherence to developing computational mechanisms for understanding natural language text, and argued that the principles involved arise from a relatively small number of logical principles. He also suggested that the tendency

---

[3] Linguists have distinguished between *coherence* and *cohesion*; the latter being applied to describe more surface level associations. In the current paper I do not use this terminology. I believe the distinction is not always clear cut.

for coherence to involve co-reference does not arise because co-reference creates coherence but rather it is the other way around: effective communication depends on certain types of progression such as elaboration and clarification, and these types of coherent progression tend to involve co-reference.

Whether or not coherence produces or is produced by co-reference does not seem to bear directly upon the current analysis. In fact, I believe that the dependency runs in both directions. But if we take the example of a particular case of pronoun resolution, such as a WS, we already know that there must be co-reference, since the pronoun must refer to something referred to elsewhere in the text. Then by appealing to the Gricean principle of non-ambiguity we can assume that there must be some principle by which the reference of the pronoun can be determined. The following example from [16] is a good example of Hobbs' approach and also a good WS example:

- John can open Bill's safe. He knows the combination.

That 'he' refers to 'John' can be explained on the basis of a principle of *elaboration*, since knowing its combination can be regarded as a more detailed explanation of being able to open a safe. This is an example general class of elaborations in which an agent's ability to do something is explained in terms of them having some kind of knowledge.

Now consider:

- John can open Bill's safe. He will have to get the combination changed soon.

Hobbs says that this is an example of a 'causal coherence relation' which depends upon knowledge of the purpose of a safe and the purpose of a combination. So it seems that Hobbs' view is that, although his relatively simple coherence relations will work in many cases, coherence may also be dependent on much more complex knowledge.

2.1. The Approach of Kehler et al.

Despite **WS1** being the original WS example and also being one for which the implied pronoun resolutions are fairly clear, explicit explanations of the principles behind its resolution are scarce in the literature. As far as I am aware, the most detailed analysis of this specific case is that of Kehler *et al.* [18]. Specifically, they say "Oversimplifying a bit, we encode the world knowledge necessary to establish explanation for **WS1** within a single axiom." The axiom they give is:

$$Fear(x,v) \wedge Advocate(y,v) \wedge Enable\_to\_cause(z,y,v) \rightsquigarrow Refuse(x,y,z) \qquad (1)$$

To clarify this they state that the implication relationship ($\rightsquigarrow$) means that the formula on the right 'plausibly follows from' those on the left.[4] They proceed to explain how (1) is sufficient to deal with both **WS1a** and **WS1b** by means of abductive inference. The idea is that when interpreting '*P because Q*', we try to match *P* with the consequence of some plausible implication an then to match *Q* with one of the antecedents of this implication. Hence, with **WS1** this would work as follows:

The first clause in both versions of **WS1** is of the form:

---

[4] This is stated in [18], but I have replaced the symbol $\rightarrow$ in their formula with $\rightsquigarrow$, to avoid possible confusion with a material implication.

- *Refuse*(*councillors*, *demonstrators*, *permit*),

so an explanation of this can be given by an instantiation of (1), with the variable assignment $x = councillors$, $y = demonstrators$ and $z = permit$. Using this assignment, the clauses on the left would be instantiated as:

- $Fear(councillors, violence)$,
- $Advocate(demonstrators, violence)$,
- $Enable\_to\_cause(permit, demonstrators, violence)$.

Thus, since **WS1a** contains $Fear(they, violence)$, we can match 'they' in this sentence to 'the councillors' and in **WS1b** we have $Advocate(they, violence)$, so in this case 'they' matches 'the demonstrators'.

I concur with many aspects of this analysis. (1) is a reasonable, albeit fairly coarse grained, representation of a general principle by which **WS1** may be resolved. It is indeed the case that, if an agent fears some outcome advocated by another agent, which would require some item to achieve that outcome, then that provides an explanation of why the first agent would prevent the second agent acquiring the item. I also agree that a sentence of the form '*P because Q*' can only make sense if Q can play a part in some possible explanation of P. Furthermore, if in such a sentence Q contains a pronoun then *P because* Q' should make sense, where Q' is formed by replacing the pronoun in Q with some proper noun or noun phrase occurring elsewhere in the sentence.

One could find various respects in which (1) may need refinement or elaboration. One could also complain that it is infeasible that one could formalise all the principles required to resolve all of the huge range of potential WS examples that could be devised. Neither of these seems to be a decisive argument against the use of principles of similar form to (1). However, I believe that there is another major problem facing this approach.

2.1.1. The Problem of Identifying the Appropriate Principle

The critical problem is how to identify the appropriate principle to apply to a particular WS example. There are countless reasons why one agent or group would deny something to another agent or group. Hence, any set of principles sufficient to handle a wide range of WS examples would contain many plausible inference rules with $Refuse(x, y, z)$ as the consequent. When we apply the induction rule we also make use of additional information such as $Fear(they, violence)$ to find matching explanation, and this will narrow down the choice of applicable rules. But can we expect this matching to narrow down the possibilities sufficiently to identify a single correct rule? I believe not. Or at least not with a rules that is as coarse grained as (1).

We may think that we can find the appropriate rule by means of the additional information given in the WS example. However, note that while (1) contains the predicates $Fear(x, v)$ and $Advocate(y, v)$ each of **WS1a** and **WS1b** mention only one of the corresponding relationships, so supposition that (1) explains the situation relies on inductive inference that the other also holds. But it does not seem reasonable that from knowing only $Refuse(x, y, z)$ and $Fear(x, v)$ we can deduce $Advocate(y, v)$. Consider this variation:

- The councillors refused the demonstrators a permit because they feared that pickpockets would take advantage of a crowd of unsuspecting middle class do-gooders milling around in the town square. It would be a nightmare for the local police.

In this case it is clear that the demonstrators would not be advocating the outcome that the councillors fear. And we can easily imagine scenarios involving permits and fear where a variety of different rules operate. For example:

- The councilmen gave the racketeers a permit because they feared blackmail.
- The psychologists refused the patients a certificate of mental health because they feared leaving their own house.

We have not yet established how the relationship $Enable\_to\_cause(z, y, v)$ that occurs in (1) might be used to guide selection of this rule. As we saw above by matching the relationships explicitly stated in **WS1a** in terms of the verbs 'refuse' and 'fear', to the rule (1) we get $Enable\_to\_cause(permit, demonstrators, violence)$. From the explanation given by Kehler *et al.* [18] it seems that they consider this relationship to be a further product the inductive inference although it is not necessary to resolve the pronoun 'they', which can be determined from $Fear(councillors, violence)$ alone. However, since, as I have pointed out, the presence of the relationships expressed in terms of 'refuse' and 'fear' does not seem sufficient to guarantee that the rule is appropriate, it could be argued that recognition of the relationship $Enable\_to\_cause(permit, demonstrators, violence)$ also plays a key role in identifying that the rule (1) is appropriate for this case.

But, since $Enable\_to\_cause(permit, demonstrators, violence)$ is not explicit in **WS1a**, we still face the problem of where would this come from. Nevertheless, it is plausible to argue that this kind of relationship could be part of background knowledge which must be employed in conjunction with explanatory rules such as (1) and includes information such as causal relationships that can occur involving particular types of agent and object. Indeed this seems to fit very well with Hobbs's suggestion that coherence principles based on causal relationships (such as between a safe and its combination) need to be specified as background knowledge. Hence, in addition to the rule (1) our theory would also contain:

$$Enable\_to\_cause(permit, demonstrators, violence) \tag{2}$$

This idea seems to me to be along the right lines. However, it is not without further problems. Suppose we have a knowledge base containing instances of the *Enable_to_cause* relation. It would contain cases such as *Enable_to_cause*(*knife*, *idiot*, *death*). Could such a knowledge base ever be complete or accurate? It seems highly unlikely that one could cover all possible cases of causal enablement without massive over generalisation. In the case of the particular relationship *Enable_to_cause*(*permit*, *demonstrators*, *violence*), one cannot assume that this is always relevant, even when interpreting sentences that mention the concepts 'permit', 'demonstrators' and 'violence'. For example, consider the following case:

- The demonstrators were protesting that the councillors had approved a permit for the knife throwing festival because they feared violence.

We need to recognise that the applicability of *Enable_to_cause(permit, demonstrators, violence)* to a particular situation depends on several further assumptions about the relationship between the three elements. Most obviously, it depends on the assumption that the permit relates to the demonstration being planned by the demonstrators and also that violence may arise from the demonstration. I will argue in the rest of this paper that such assumptions are plausible and often necessary for the interpretation of natural languages. However, I shall claim that such connections do not in most cases arise from general relationships of the form of (2). Instead they arise primarily from a kind of coherence property, which, as far as I am aware, has not been emphasised in any previous work on coherence. The type of coherence to which I wish to draw attention results from a principle *identification of implied entities*. This is a principle that we apply pervasively but largely unconsciously in our interpretation of natural language. In the remainder of the paper I shall attempt to explain how this works and why it is so powerful.

3. A 'De-Coherent' Interlude

So you have probably heard the news: "The city councillors refused the demonstrators a permit because they feared violence." Let me elaborate further:

> The councillors of Bolzano had been approached by a party of Dutch clog makers, who were on their way to Lagos to hold a demonstration against import tariffs on hand-painted clogs. The prospective demonstrators were from Leiden but were requesting a permit on behalf of a group of farmers from the neighbouring town of Zoetermeer, from whom they often bought tulips, and who (because of floral oversupply in Holand) wished to relocate to Colombia to set up a tulip plantation. The opportunity to carry this out was made possible by the 'Los valientes pueden crecer' (the brave may grow) initiative, a scheme by which city councils of other countries could issue agricultural licences to farmers wishing to establish cultivation in Colombia. The reason that the Lieden clog makers had chosen to stop in Bolzano on their way to Lagos was primarily to visit some cheese makers with whom they had a trading relationship and from whom they had learned that the Bolzano council had recently issued a permit to allow some wine growers from Merano to set up a vineyard near Medellín. One of the Zoetermeer farmers had heard about this when making a tulip delivery to Lieden; and since the Bolzano council were clearly familiar with the scheme and the required paperwork, it made perfect sense for the Lieden clog makers to apply for the permit on behalf of the Zoetermeer farmers during their stay in Bolzano.
>
> However, a condition of the 'brave may grow' scheme was that such permits could only be given to prospective farmers who would not only produce nourishing food but also be capable of defending the land they would be allocated, against brutal and heavily armed drug cartels (who preferred the cultivation of cocaine to other vegetation). When the Bolzano councillors interviewed the Dutch delegation, they found them to be of very different character from the mountain toughened South Tyrolean wine growers, whose permit they had previously approved. One clog maker let slip that she and her fellow artisans were greatly worried that they might face violent aggression from the Nigerian authorities during their planned demonstration in Lagos. And, since the Dutch clog-makers seemed so afraid of potential violence from the Nigerian police, the Bolzano councillors judged that their tulip growing compatriots were likely to be of similarly meek disposition, and would be no match for Colombian drug lords. So the councillors refused to approve the permit because the demonstrators feared violence.

If you had thought it was the city councillors that feared violence, you were mistaken. Why did you think that? Probably the reason was that you applied a *coherence preference* in your interpretation of (1). You assumed that the city councillors were councillors of the *same* city in which the demonstrators planned their protest. And you assumed that the permit was a permit for these *same* demonstrators to hold a demonstration in that *same* city. But no, the situation involved several different locations, and the requested permit had no direct connection to the planned demonstration.

I am *not* arguing that **WS1a** is genuinely ambiguous. I believe that the ‘correct’ pronoun resolution for **WS1a**, when seen on its own is that the ‘they’ refers to the councillors. My counter-interpretation is very complex and artificial. My reason for constructing it was not primarily to show that the pronoun could be interpreted differently but rather to highlight the strength and pervasiveness of coherence principles that condition our interpretation of natural language. I describe my interpretation as *decoherent* because it deactivates usual coherence conventions by statements that negate identities between entities that would otherwise be assumed to be the same.

4. The Logic of ‘Because’

Many of the Winograd Schemas are of the form ‘ϕ because ψ’. The meaning of ‘because’ is somewhat difficult to define. It is generally agreed that ‘ϕ because ψ’ implies ‘ϕ and ψ’. However, it is also clear that ‘ϕ because ψ’ says more than just the truth functional conjunction. Informally, we may that ‘ϕ because ψ’ is true whenever both ϕ and ψ are true and ψ gives an *explanation* of ϕ. Schnieder [19] presents a logical calculus for the ‘because’ connective using Natural Deduction style rules. The intuition underlying that system is also that ‘ϕ because ψ’ holds when ψ explains ϕ. However, the rules of Schnieder’s calculus are limited to cases where the form of explanation is itself purely logical. For example, one rule says that from ϕ we can derive ‘$(\phi \vee \psi)$ *because* ϕ’.

Let us analyse ‘because’ in terms of what it means for one statement to provide an explanation for another. Consider a statement ‘x did A because P’, where A is some voluntary action performed by x. In such a case, this only makes sense if P gives some reason that explains why x would choose to do A. The explaining statement can be of many forms and can refer to a very wide range of possible factors that could motivate x to perform A. We may distinguish two broad categories of explaining statement:

- those that refer to some mental property of x (such as a belief, desire or intention),
- those that refer to some claimed fact about the world (including possible future occurrences and also the actions or possible actions of other agents in the world).

For present purposes, I shall consider only the second type of explanation. In such a case, the claimed fact P is proposed as an explanation of x’s action A, without any explanation of why P would motivate this action. Thus we must fall back on an implicit, generic explanation of how a fact would motivate a action. I suggest the following:

- On the basis of P, together with other background and contextual knowledge, it is possible to reason that either:
    - ∗ doing A will have an outcome that is good for x;
    - ∗ *or*, **not** doing A may lead to a state that is bad for x;

Here, what is ‘good’ or ‘bad’ for x should be interpreted very generally. As well as material benefits or adversities, it includes conditions of status and obligation. Thus, the outcome of fulfilling an obligation or duty would be considered good and of failing to fulfil an obligation or duty would be bad.

4.1. Logical Properties of an 'Explains' Connective

I define the notation $\phi \rightsquigarrow \psi$ to means that 'ϕ can provide an explanation for ψ'.[5] From consideration of Schnieder's account of 'because' and also the specific requirements for resolving **WS1**, I propose that the $\rightsquigarrow$ connective provides the following minimal principles of inference:

Contingent Entailment: $\phi \rightsquigarrow \psi$ if ($\phi \vdash \psi$ and $\nvdash \neg\phi$ and $\nvdash \psi$)

Lexical Semantic Implication: $\phi \rightsquigarrow \psi$ if ψ can be obtained from ϕ by applying axioms expressing semantic properties and relationships among vocabulary terms.

Transitivity: If $\phi \rightsquigarrow \psi$ and $\psi \rightsquigarrow \xi$, then $\phi \rightsquigarrow \xi$.

5. A Partial 'Formalised' Solution

I now present an account of the inference patterns that underlie the resolution of **WS1a**. The presentation is 'formalised' in a weak sense. Axioms are suggested that are intended to express the logical form of valid inferences but a proof system and semantics are not given. The following notations will be used:

- $\phi \rightsquigarrow \psi$ means that 'ϕ can provide an explanation for ψ' and follows the principles stated in the previous Section.
- The variables e range over *possible events*, that is potential occurrences that may or may not actually happen.
- $\mathsf{Occurs}(e)$ means that the possible event e actually occurs.
- $\mathbf{B}_a\,\phi$ means that agent a believes that proposition ϕ is true.
- $\mathit{Good\_for}(\phi,a)$ and $\mathit{Bad\_for}(\phi,a)$ mean. respectively that ϕ being true is good for or bad for agent a.
- All un-subscripted single letter free variables (e.g. a, x, e) are taken as universally quantified with wide scope.
- Subscripted single letter free variables (e.g. a_1, e_1) are Skolem constants (i.e. existentially quantified with wide scope).

I also define the conditions where a possible event would be good (or bad) for an agent as follows:

$$\mathit{Good\_for}(e,a) \equiv_{def} (\mathsf{Occurs}(e) \rightarrow \phi) \wedge \mathit{Good\_for}(\phi,a) \tag{3}$$

$$\mathit{Bad\_for}(e,a) \equiv_{def} (\mathsf{Occurs}(e) \rightarrow \phi) \wedge \mathit{Bad\_for}(\phi,a) \tag{4}$$

Note that the formulation I use here does not include any explicit represent of time and temporal relationships. These would certainly be necessary for a more generally applicable framework, and the scenario described in **WS1** does imply certain temporal relationships. However, it seems that temporal relationships do not play an essential part in the reasoning required to justify the pronoun resolution.

---

[5] So the symbol has very similar, but slightly different meaning from how it was used in my earlier explanation of the formulation of Kehler *et al.* [18].

5.1. *Instantiations of* **WS1a**

The following formulae are representations of respectively the ‘correct’ and ‘incorrect’ versions of **WS1a** with for each of the possible candidates being substituted for ‘they’:

$$Fear(councillors, violence) \leadsto Refuse(councillors, demonstrators, permit) \tag{5}$$

$$Fear(demonstrators, violence) \leadsto Refuse(councillors, demonstrators, permit) \tag{6}$$

To demonstrate a solution for **WS1a** we need to show that from some intuitive and general principles we can derive (5) but cannot derive (6).

Note that I am ignoring any quantification that may be implicit in the noun phrases ‘the demonstrators’, ‘the councillors’, ‘a permit’. Although, quantification is of course often very important in explaining reasoning, I believe that in this particular example it does not play a significant part and that trying to account for it would unnecessarily complicate the exposition.

5.2. *Explanation of a ‘Preventing’ Action*

If an agent believes that the occurrence of an event implies a possible state that is bad for the agent then that provides an explanation why the agent would try to prevent the event.

$$\mathbf{B}_a Bad\_for(e, a) \leadsto Try\_to\_prevent(a, e) \tag{7}$$

For an agent to fear violence means that the agent believes there is a possible event such that if it occurs it will have result that is bad for the agent. This is a semantic property of the verb ‘fear’:[6]

$$Fear(a, violence) \rightarrow \mathbf{B}_a \exists e[Violent(e) \wedge Bad\_for(e, a)] \tag{8}$$

Let us substitute *councillors* for a. Then, under the assumption that the domain of possible events includes all events that anyone might believe could exist, we can Skolemise the $\exists e$ and replace with an arbitrary event constant e_1. We then get:

$$Fear(councillors, violence) \rightarrow \mathbf{B}_{councillors}\, Bad\_for(e_1, councillors) \tag{9}$$

Since (9) expresses a purely semantic implication, it can be considered as an explanation, so entails:

$$Fear(councillors, violence) \leadsto \mathbf{B}_{councillors}\, Bad\_for(e_1, councillors) \tag{10}$$

Then, combining (10) and (7) by instantiating the variable in 7) and using the transitivity of ‘$\leadsto$’ gives:

$$Fear(councillors, violence) \leadsto Try\_to\_prevent(councillors, e_1) \tag{11}$$

5.2.1. *Refusing a Permit is a Way of Preventing an Event*

We need establish that refusing a permit is a way of trying to prevent an event. In order to do this we need to examine how a permit is related to various agents and possible events. Some consideration will reveal that a permit is a very complex item in terms of the relationships that it involves. I suggest that, even after some simplification, a permit involves at least the following implied relationships and entities:

---

[6] In a more detailed representation the *Fear* relationship would be be an attitude towards a *future* event. An exploration of how one might define emotion concepts can be found in [20].

- an agent or institution with the power to issue or approve the permit,
- the person or group to which the permit confers permission,
- the activity or event that the permit permits,[7]
- the location where the permitted activity or event may take place,
- the time period during which the permitted activity may take place,
- the rules of eligibility of the permit.

Given that permits are such complex things, the formalisation of actions involving permits is quite tricky and could be done in various different ways, depending on how you want to decompose the actions and bundle together the related entities. Assuming that the list of relevant entities I have given is sufficient to uniquely determine a possible permit,[8] we can represent the permit as a functional term, with the meaning that the term denotes a permit function that is determined by its relationship to these entities. For, example Sky City council may have the power to authorise a permit for Janet Jones to use a jet-pack within designated areas of Sky City during daylight hours and subject to specified restrictions. I this case $\mathit{permit}(scc, jj, ujp, da, dl, r)$ would denote a potential permit involving the designated entities (abbreviated by initials) fulfilling the roles described above. In many cases, we will not know all the entities relating to the permit (e.g. we may know that Janet has a jet-pack licence but not who issued the permit or what areas or times it is valid for). In such cases we can simply replace the names of unknown entities with existentially quantified variables.

We now need to clarify and define what is meant by 'x refused y a permit'. This has considerable underlying complexity. The issuing of a permit might involve several stages, and be subject to different kinds of refusal. Also, in some cases, one might request a permit on behalf of another person (e.g. a parent on behalf of a child). For present purposes I consider a simplified but typical case, where an agent or group applies for a permit relating to that same agent or group. I will interpret the relationship $\mathit{Refuse}(x, y, \mathit{permit})$ as a concise way of stating that an event occurs where x refuses to authorise a permit (regarding which they have authority):

$$\begin{aligned}&\mathit{Refuse}(x, y, \mathit{permit}) \leftrightarrow \\ &\qquad \exists e \exists l \exists d \exists r[\ \mathsf{Occurs}(\mathit{refuse}(x, \mathit{authorise}(x, \mathit{permit}(x, y, e, loc, dur, rules))))]\end{aligned} \tag{12}$$

If we now consider possible explanations of why a permit might be refused, it is apparent that wanting to prevent the event that it would permit is a good general explanation for such a refusal. We can formalise this idea with the following axiom:

$$\mathit{Try\_to\_prevent}(x, e) \rightsquigarrow \mathsf{Occurs}(\mathit{refuse}(x, \mathit{authorise}((x,\ \mathit{permit}(x, c, e, l, d, r))) \tag{13}$$

Hence, in a specific case of **WS1a**, where the councillors refuse the demonstrators a permit, this justifies the explanation:

$$\mathit{Try\_to\_prevent}(\mathit{councillors}, e_2) \rightsquigarrow \mathit{Refuse}(\mathit{councillors}, \mathit{demonstrators}, \mathit{permit}) \tag{14}$$

---

[7] Strictly speaking, permits will be valid for some *type* of event rather than a specific event occurrence (even in the case of a permit for a one-off event it would apply to many different ways in which a particular event could occur). While this is a significant ontological distinction, it does not appear to be critical for the **WS1** example, so I shall assume that a permit is in relation to an individual (possibly non-continuous) event entity.

[8] I am aware that we now have possible objects as well as possible events being referred to, but this seems to be necessary for interpreting the refusal of a permit.

It is important to note that e_2 refers to some particular but unspecified event. It is a Skolem constant arising from the existentially quantified implicit event for which the permit is valid.

Now from (11) and (14) together with the transitivity of '$\rightsquigarrow$' we would like to derive:

$$\mathit{Fear}(\mathit{councillors},\mathit{violence}) \rightsquigarrow \mathit{Refuse}(\mathit{councillors},\mathit{demonstrators},\mathit{permit}) \tag{15}$$

However, there is a major problem. Our analysis of **WS1a** reveals implicit references to two events: the event that the councillors fear, and the event that is permitted by the permit. We can only infer (15) if these are the same event (or at least must occur together — we could regard the violence as pertaining to an event that is only part of the whole demonstration event). I believe such an assumption is necessary for the resolution of **WS1a** and exemplifies a key mechanism for enabling natural language understanding.

5.2.2. A Default Rule for Entity Identification

One would like to have a general way of establishing identity between different entities referenced either explicitly of implicitly. In the spirit of *Ockham's razor* one can formulate a general rule of default inferences of the following form:

$$\frac{\mathcal{O},\mathcal{K},\, I \,\vdash\, \exists x \exists y[\Phi(x,y)] \qquad \text{and} \qquad \mathcal{O},\mathcal{K},\, I,\, \exists x[\Phi(x,x)] \,\nvdash\, \bot}{\mathcal{O},\ \mathcal{K},\ I \,\vdash\, \exists x[\Phi(x,x)]}\,\textbf{DEI}$$

This says that, if, from ontology $\mathcal{O}$, with background knowledge $\mathcal{K}$ and some given information I (e.g. a description of some scenario), we can infer the existence of two entities satisfying some relation Φ, and it is also consistent with $\mathcal{O}$ and I that these entities may be the same, then we can (by default) infer that they are the same. For this to give reasonable inferences we would need to ensure that any semantic constraints implied by the context of x and y in Φ are enforced by $\mathcal{O}$ and $\mathcal{K}$. Even with this proviso, the rule may be too strong and it may be very difficult to determine exactly what knowledge should be incorporated within $\mathcal{K}$. Nevertheless, if applied with suitable caution **DEI** may be a useful form of inference for natural language interpretation.

5.2.3. Further Justifications of the Inference

As justification for the pronoun resolution we may seek to find a reason why the councillors would consider a violent demonstration to be bad for them:

- Every city council has responsibility for a city.
- If an agent or organisation a is responsible for some thing x, then x being in a bad condition is bad for a.
- If a violent event occurs in a location it is bad for that location.

These conditions could be represented formally as:

$$\forall x[\mathit{Councilors}(x) \rightarrow \exists y[\mathit{City}(y) \wedge \mathit{Has\_responsibility\_for}(x,y)]\ ] \tag{16}$$

$$\mathsf{Occurs}(e) \wedge \mathsf{Loc}(e,\mathit{loc}) \wedge \mathit{Violent}(e) \rightsquigarrow \mathit{Bad\_condition}(\mathit{loc}) \tag{17}$$

$$\mathit{Has\_Responsibility\_for}(a,x) \rightarrow \mathit{Bad\_for}(\mathit{Bad\_condition}(x),a) \tag{18}$$

Again the use of this knowledge in interpreting **WS1a** relies on identification of implied entities: we must assume that the demonstration that the demonstrators are planning will take place in the *same* city that the councillors are responsible for and also that the requested permit is a for a demonstration to take place in that *same city*.

Another reason why we might want to justify the interpretation of **WS1a** is that demonstrations are the kind of event that is likely to turn violent. However, this does not seem to be necessary for the pronoun resolution. I believe that in the following cases we would still normally interpret ‘they’ as referring to the councillors:

- The councillors refused the funfair organisers a permit because they feared violence.
- The councillors refused the funfair organisers a permit because they feared Joe Carson would turn up.

It is clear that fear of violence can influence agents’ actions in many ways. A couple of other examples that illustrate this diversity are:

- The organisers of the demonstration decided not to apply for a permit, because they feared the event would turn violent. In fact they actually advocated violence, so they didn’t want their names on a permit application form.
- The samurai offered to protect the villagers because they feared violence.

6. Conclusions

My investigation of **WS1** has only been partially successful. Despite fairly elaborate semantic analysis, the explanation of the required inferences still has several loose ends. I think it did shed light on what the problems are and how one might go about addressing them, but the methodology can certainly be called into question with respect to its generalisability. KRR approaches require huge amounts of detailed representation of both lexical semantics and world knowledge, so expanding such analysis to everything that could be described in natural language is daunting and may seem infeasible. However, in [12] I and my collaborator have investigated the coverage of a set of rules relating to the verb ‘thank’ and found that these rules could account for approximately 0.4% of WS examples in the large WinoGrande set [9]. Although this is a small fraction, it does lend credibility to the idea that one could incrementally build up to much greater coverage by adding knowledge domains in a modular way.

One should also consider generality in the types semantic structures and rules that have been identified. Here I believe a strong case can be made. Rules expressing general forms of plausible explanation, such as motivations for an agent carrying out an action seem (as was also observed in [12]) to be transferable to a wide range of scenarios and to many WSC examples. One can also argue that axiomatising the notion of ‘reasonable explanation’ may, for many purposes, be more effective than trying to give a logical theory of the philosophically problematic concept of *causality*. The need to identify implied entities is is especially salient vor **WS1**. However, my informal examination of WS examples suggestst that around 75% also depend on some kind of entity resolution (in addition to the pronoun resolution) though it is often of a less distinctive form (a very typical case is where two parts of a sentence refer to two aspects of an event).

The current analysis consists of a rather *ad hoc* combination of logical syntax with no explicit semantics. My aim was narrowed to finding a plausible path of inference to account for just one example, but ideally we would prefer a general purpose logical language with a precise syntax and semantics. Of course, many such frameworks have already been proposed and it is apparent that they have features that address some issues raised in the current paper. For instance Minsky's *Frames* [21] and Schanks' *Scripts* [22]. group together concepts and relationships associated with a particular type of object, situation or event. Hence they would support the representation of 'permit' and its dependent entities in a way that is similar to what I proposed. The *ontologies* used in modern advanced information systems fulfil a similar role to information organisation to structures such as Frames. But, whereas Frames and Scripts typically specify conceptual structures focused around particular types of object or situation, ontology languages such as OWL are more oriented towards specifying abstract relationships between concepts, such as subsumption hierarchies. Both these forms of knowledge appear to be essential to finding coherent interpretations of natural language and especially the problem of identifying implied entities and using them to glue together the parts of a sentence. Frame type organisation of knowledge is good for identifying the auxiliary relationships and entities that surround every concept in every description, and ontologies can specify the categorial constraints on types of entity and possible relations between them that are required for establishing connections between these implied entities.

The method of investigation carried out in the current paper is likely to be significantly enhanced and generalised by incorporating insights and theories from other research that I have more recently become aware of. In particular, work in formal linguistics has developed frameworks such as Dynamic Semantics and Segmented Discourse Representation Theory [23], that provide logical representations that can capture more complex interplay of language features than is possible in the traditional, more static first-order logic. Also the notion of *bridging anaphora* [24] has long been known in the field of language processing overlaps substantially with my idea of resolving identities of implied entities, and algorithms have been developed for finding bridging links [25].

References

[1] Levesque HJ, Davis E, Morgenstern L. The Winograd Schema Challenge. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference; 2012. .

[2] Winograd T. Understanding natural language. Cognitive psychology. 1972;3(1):1-191.

[3] Levesque HJ, Davis E, Morgenstern L. The Winograd Schema Challenge. In: Brewka G, Eiter T, McIlraith SA, editors. Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012. AAAI Press; 2012. .

[4] Rahman A, Ng V. Resolving Complex Cases of Definite Pronouns: The Winograd Schema Challenge. In: EMNLP-CoNLL; 2012. .

[5] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:181004805[csCL]. 2018.

[6] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:190711692[csCL]. 2019.

[7] Trichelair P, Emami A, Trischler A, Suleman K, Cheung JCK. How Reasonable are Common-Sense Reasoning Tasks: A Case-Study on the Winograd Schema Challenge and SWAG. arXiv:181101778[csLG]. 2018.

[8] Kocijan V, Cretu AM, Camburu OM, Yordanov Y, Lukasiewicz T. A Surprisingly Robust Trick for Winograd Schema Challenge. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics; 2019. p. 48374842.

[9] Sakaguchi K, Bras RL, Bhagavatula C, Choi Y. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. In: AAAI-20; 2020. .

[10] Emami A, Suleman K, Trischler A, Cheung JCK. An Analysis of Dataset Overlap on Winograd-Style Tasks. In: Proceedings of the 28th International Conference on Computational Linguistics; 2020. p. 5855-65.

[11] Ettinger A. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. Transactions of the Association for Computational Linguistics. 2020;8:34-48.

[12] Hong SJ, Bennett B. Tackling domain-specific winograd schemas with knowledge-based reasoning and machine learning. In: Proceedings of the 3rd Conference on Language, Data and Knowledge (LDK); 2021. .

[13] Bailey D, Harrison A, Lierler Y, Lifschitz V, Michael J. The Winograd Schema Challenge and Reasoning about Correlation. In: 2015 AAAI Spring Symposium Series. USA; 2015. .

[14] Sharma A, Vo NH, Aditya S, Baral C. Towards Addressing the Winograd Schema Challenge - Building and Using a Semantic Parser and a Knowledge Hunting Module. In: IJCAI 2015; 2015. p. 1319-25.

[15] Grice HP. Logic and conversation. In: Speech acts. Brill; 1975. p. 41-58.

[16] Hobbs JR. Coherence and coreference. Cognitive science. 1979;3(1):67-90.

[17] Lapata M, Barzilay R. Automatic evaluation of text coherence: Models and representations. In: IJCAI. vol. 5; 2005. p. 1085-90.

[18] Kehler A, Kertz L, Rohde H, Elman JL. Coherence and coreference revisited. Journal of semantics. 2008;25(1):1-44.

[19] Schnieder B. A logic for 'because'. The Review of Symbolic Logic. 2011;4(3):445-65.

[20] Wierzbicka A. Defining emotion concepts. Cognitive science. 1992;16(4):539-81.

[21] Minsky M. A framework for representing knowledge. MIT-AI Laboratory; 1974. Memo 306.

[22] Schank RC, Abelson RP. Scripts, Plans, and Knowledge. In: Proceedings of the 4th International Joint Conference on Artificial Intelligence - Volume 1. IJCAI'75. Morgan Kaufmann; 1975. p. 151-7.

[23] Asher N, Lascarides A. Logics of Conversation. Cambridge University Press; 2003.

[24] Clark HH. Bridging. In: Theoretical issues in natural language processing; 1975. .

[25] Hou Y, Markert K, Strube M. A rule-based system for unrestricted bridging resolution: Recognizing bridging anaphora and finding links to antecedents. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 2082-93.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210370

Towards an Ontology of Representation

Riichiro Mizoguchi[a] and Stefano Borgo[b]
[a] *Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan*
E-mail: mizo@jaist.ac.jp
[b] *Laboratory for Applied Ontology, Institute of Cognitive Sciences and Technologies (ISTC-CNR), Trento, Italy*
E-mail: stefano.borgo@cnr.it

Abstract. In philosophy information is mainly discussed along with the notion of aboutness. In more practical communities, information is mainly addressed together with notions like data and knowledge. This paper proposes a different approach. We look at information (and related concepts) as roles played by representations. This view implies that the notion of representation is central for any ontological analysis of information and related concepts. The paper provides arguments for this new stand and discusses an ontological model of representation based on the systematic distinction between form and content. The broadness and flexibility of the proposed model is shown by discussing a list of variegated representation entities from music to procedure, from novel to painting. The paper also investigates the role of letters (characters) in natural language expressions, which turns out to be quite complex.

Keywords. Representation, form, content, representing thing, specification, information

1. Introduction

Information is usually defined in ontology as a content-bearing entity that exists in reality [1]. This status of content-bearing follows from the assumption that the generated information is about what is observed, and the status of *being about something* is a problematic. Given this view, it is natural to take as a must for an ontological investigation of information to address the ontological status of aboutness [2][5], a notion that encounters important difficulties including the problem of 'fake' information (information which do not correspond to how things are in reality). One could argue that fake information is an epistemological issue and, if one takes the realist approach, this puts it out of scope of applied ontology. Yet, many consider this the sign of the inadequacy of our understanding of aboutness, and, in turn, of what information is.

In this paper we propose to take a different path. It claims that to understand information one must first understand representation. In other words, it says that in ontology representation is more fundamental than information. More cautiously, we observe that a theory of information presupposes a theory of representation and, thus, the latter must be developed to satisfactorily investigate the former. To support this view, one should note that information can be seen as a role played by a representation in the context of an event "*to inform*" (see Section 2). After all, information is borne in some type of representation, and some would even go as far as claiming that information itself must be borne to exist [1].[1]

---

[1] An informal version of this approach was published in [3] [4].

One can start from apparently naïve questions to expose the core of this issue. Consider, e.g., this question: "What is an instance of (a piece of) music?" Two answers come right to mind: it is a sequence of sounds (an event), and it is the score written by the composer or a copy of it. Music is a useful example in the discussion of representation because it seems immediate to the layman but formally it is hard to model. In the first case, namely, for music as the sound generated by instruments and players, a piece of music has nothing to do with representation. However, one could observe, the players performing the music follows a musical score (physically reading it or just remembering it by heart), and that musical score is a representation.[2] From this observation, we can now deepen the initial answers about music. A 'piece of music' could refer to four distinct ontological entities:

(1) What the composer coded in musical language (the specification of the sound sequences the composer wants to characterize)
(2) The musical score (a text that has an associated semantics)
(3) The performance of the player (an activity)
(4) The sound generated by the performance (an event)

Let us now analyze the ontological status of these entities and the relationships between them. First, these four entities have clear relationships: the set of constraints/conditions (1) stated by the composer are captured by the semantics of the musical score (2) whose purpose is to make them explicit and shareable. The performance (3) is a realization process of the set of constraints such that the sound generated (4) in the performance is a realization of the sound patterns specified by them. Since the musical score, entity (2), is a representation in the ontological sense, our analysis will center around it. The reader should keep in mind that this is only an example and that our aim is not limited to music scores. Indeed, we aim to develop an ontology of representation that equally applies to *procedures, recipes, dramas, novels, poems, paintings, car models*, and the like. To be more precise, the notion of representation discussed in this paper covers the notions of *depiction* and of *description*; it does not include other representation meanings like "being appointed to act/speak for", "standing for" and so on. Briefly put, we investigate ontological representation objects, not role-based representation objects as discussed, e.g., in semiotics.

We start with the assumption that there are language(s) and that they are endowed with semantics. These are the languages we care about. We also assume that these languages exist in time but not in space. They are created but not material in ontological argot. That is, we look at languages as set of rules (syntactical and semantical), not as sets of material marks. (One can be more demanding, for instance asking that these rules must always have a material, digital or neuronal support, but this issue is orthogonal to our work.) Note that our focus is on ontological aspects, not linguistic ones. Thus, we assume that these languages are not affected by ambiguity or, for what it matters, that any possible ambiguity has been already resolved when we discuss representations. Language can be symbolic, natural, analogical, iconic etc. including a mixture of these.

Given such a language, we take an *ontological representation object* to be an object that is necessarily composed by two ontological parts: an expression in that language (as said, not a material rendering of the expression), which we call *form*, and the meaning of that expression according to the language's semantics, which we call *content*. It follows that an ontological representation object exists in time but not in space,

---

[2] Improvisation in music, like in jazz sessions, is of course a separate issue and does not fall within the scope of this paper.

as it is not a concrete entity not even in a parasitic sense like holes, and is independent of the existence of an agent that can understand or even recognize the object. To be precise, the relationship between a representation and the agents that understand, recognize or even create it is not part of the analysis we carry out (on the relationship between a representation and the human brain see, for instance, [1,2]). The relationship between form and content is called *encoding*. Here we do not discuss the specific characteristics of an encoding relation; it suffices to observe that it is usually based on (a combination of) the following: social conventions [7], private conventions (e.g., individual decisions of an agent), and similarity (e.g., the topological similarity between a subway map and a railroad network). This paper is about ontological representation objects whose encoding is based on social conventions. To avoid a possible misunderstanding, we clarify that *encryption* (as in cyphertext) is not a type of encoding as discussed in this paper. Indeed, an encryption is a relationship between forms of two languages (one of which may not even satisfy our definition of language since it does not need to have a semantics). From now on, we will use the term *representation* for ontological representation object.

Note that our notion of representation does not exclude “representation-of”. It subsumes “representation-of”. Consider a representation corresponding to sentence “John Smith is the third author of the paper”. The representation, differently from the sentence, does not depend on the referred person John Smith, nor on what “the paper” is. Our representation theory does not aim to establish whether a representation is true or not, nor if there exists something in the world to which it refers, etc. Even representations that, taken as linguistic expressions, one may claim do not have a reference (“The French emperor is bald”, “The sun rises from the West in Japan”) are representations: they have form and content, and this is independent of what they are about (if anything). For another example, consider a procedure. A procedure is a representation; thus, it has a form and a content. It is important to understand that the content of a procedure is a specification for actions. A procedure does not refer to a/the/all/any procedure realization, and it is not about actions. Rather, it states the conditions a set of actions should satisfy to be a realization of that procedure. One can even give a procedure that is physically or even logically impossible to realize, it is a procedure to all effects since it has the needed form and content.

The paper is organized as follows. The next section discusses data, information, and knowledge, and presents these as roles played by a representation. It follows that representation is a fundamental concept for understanding them. Section 3 discusses form and content as components of representation. It also addresses the ways a representation can be realized, namely, the *form-realization* and the *content-realization*. Section 4 discusses the concept of *representing thing*: the concretization of a representation obtained by form-realization. In section 5 a preliminary formalization of representations and letters is presented to ease the understanding of the theory and to highlight dependences across the entities. Section 6 gives an overview of the entities introduced, adds considerations on the notions of copy and of identity of representation. It also discusses a different type of content needed to properly model the form of texts, paintings, etc. where form has a special value. Section 7 compares this approach to the IAO theory and other views. Section 8 adds the concluding remarks.

2. Data, Information and Knowledge as Representations

There have been a lot of discussion about how to understand data, information and knowledge and their relationships [6] and people have tried to distinguish them primarily according to intrinsic properties and characteristics that may help to answer questions like “what is the difference between data and information?”, “what can one do when she has knowledge vs. when she has data?” These attempts have not led to consensus [6]. Informally speaking, an expression like “Tokyo is the capital of Japan” becomes data when stored in a database, information when provided to another agent (human or artificial) who does not have it, and knowledge when an agent has learned it. While an appropriate discussion would take a full paper, we notice that all the discussions about data, information and knowledge address kinds of representations. This observation suggests that a representation *becomes* data, information or knowledge according to the context in which it is considered. This leads us to the following characterization:

a) Data is a representation when participating in an event in which it is processed, i.e., when the representation is the operand of a “to process” event. For instance, a person’s name, a bare representation, is data when stored in a repository.

b) Information is a representation when participating in an event in which it is exchanged, i.e., when the representation is the operand of a “to inform” event. For instance, a person’s name, a bare representation, is information when shown by a repository to an agent. Note that the status of being information is not related to the usefulness of the representation for some purpose, indeed there is no assumption that a purpose exists.

c) Knowledge is a representation when participating in an event in which it is learned (by an agent), i.e., when the representation is the operand of a “to learn” event. For instance, a person’s name, a bare representation, is knowledge when an agent learns it (e.g., the agent can now use this representation to correctly address the person).

We acknowledge that these definitions may look limited. The stacks are high when discussing data, information, and knowledge, and any alternative view can easily look disappointing. However, at this stage we want only to highlight a general difference between the act of *processing* and the thing processed; the act of *informing* and the thing which is informed; the act of *knowing* (or *learning*) and the thing which is known (learned). While the thing processed, informed, or learned can be the same one, the acts are not. As suggested above, we look at data, information and knowledge as roles played by a representation in the context of *processing, informing* and *learning/knowing*. This view is grounded in the theory of roles presented in [10][11]. The assumption is that what people call data, information or knowledge is not intrinsically such. It follows that one needs to characterize the activities which provide a context in which these roles are defined. A first positive consequence of this choice is that it seems much easier to distinguish these acts, even though the specific understanding of each can vary from community to community. The confusing results in understanding data, information and knowledge [6] is a motivation for exploiting a different view which hopefully will turn out to be more precise and coherent. In conclusion, we have grounds to claim that representation should be investigated as an alternative foundation for the meaning of terms like data, information and knowledge.

3. On the Classification of Representations

In the previous section we said that to make sense of data, information and knowledge one has to consider several ontological entities: activities, objects, roles and representations. In this section we show that our proposal is general enough to cover the variety of representation types. To do so, we consider seven distinct types of representation, namely: *procedure, music, drama, novel, poem, handwriting (in calligraphy)* and *painting*.

In the standard understanding of these types, all instances of the first five (procedure, music, drama, novel, poem) are created because of their content. A procedure's content is a specification of actions; a (piece of) music's content is a constraint/condition on sequences of sounds; a drama and novel's contents are sequences of happenings; a poem's content is a sequence of a variety of elements like scenes, thoughts, emotions. Handwriting and painting have content in a different way[3], we can assume that the existence of these entities is primarily due to the form. We will come back to this issue below.

All these representations have **authors/creators** and **users**, and the first three also have **performers**, or so it is usually assumed. By authors, we mean agents who created the representation. By users, we mean agents who attend a performance (an event) based on the representation, for instance readers (in public reading events), audience or spectators. By performers, we mean agents that act following the representation content (thus, the public reading or recitation of a novel or poem does not count as performance and readers are not performers). For the first three, they can be human or artificial agents, musicians, and actors. The last four do not have performers. Those who at first sight seem to be performers are actually users, they are analogous to the listeners of music and spectators of drama. A poem recitation is sometimes considered a performance, it is not as we will see later. Note that users are not in the scope of our discussion because they do not characterize representations.

We said that representations are composed of two parts: form and content. By comparing different representation types, we can see that in some cases their reason to exist weights on one component more than the other. As for the first five types, the focus is clearly on the content which has only a generic dependence on the form. As to the last two, the form has a primary role to the point that the content is strictly dependent on the first: "the moral and philosophical implications in Chinese calligraphy […] are associated directly with […] the brush strokes and the way space is used, not merely derived from the general meaning of the words." (p.19, [8]). We already noticed that some representations, like music, are associated with two activity types: composing (creating) and performing. What can we say about painting? At first sight, a painter seems to be executing a composition (creation) act, as she is creating something anew as opposed to performing something according to specifications. It follows that music and painting are ontologically different representations: both have creators but only the first has performers.

Usually, one talks of *realization* as a mapping from a source to a destination, where the destination (the result of a realization process) is a concrete entity. As to the source, we limit ourselves to representations. Since a representation is composed of form and

---

[3] These two cases are sophisticated and are here treated somehow naively. They are considered mainly to show that our approach is applicable. Note that the meaning of the words or sentences in calligraphy is not important. A portrait has a content (the depicted person) which, we claim, is not important *per sé*.

content, we can have **form-realization** and content-representation. The first may exist for all kinds of representation. Regarding **content-realization**, however, it may exist for procedures, pieces of music, and dramas only. In the case of the recitation of a poem, the source is essentially the form of the representation, so such a recitation act is not a performance. Although the content of a poem becomes relevant for other reasons (like artistic quality), it does not deal with the realization itself. When the content-realization exists, the representation content is called a *specification*. On the other hand, the form of a representation admits multiple realization types starting from: **object realization** (e.g., a sentence written on a sheet of paper) and **process realization** (e.g., speaking a sentence, reciting a song's lyrics). In sum, (1) *procedure, piece of music* and *drama* have both form-realization and content-realization; (2) *novel, poem, handwriting* (in calligraphy) and *painting* have form-realization but no content realization. Furthermore, the latter are obtained by a *production activity* as described below.

Consider engineering artifacts. They have two associated and clearly distinct phases, namely the design phase and the production phase. For what concerns us, these are two types of activities, they both have an outcome but of different type. We use design and production activities in artifacts to characterize the distinction between representation and realization in our theory. From this viewpoint a musical composition is a specification (a design, thus a set of rules) written in a language suitable for music, and a sequence of sounds satisfying the specification is an execution (a production, thus an object that satisfies the design). The composer, by creating music, produces a (material) musical score which is a form-realization of the musical composition (the composer realizes the design). The production corresponds to a content-realization in which a sequence of sounds is manufactured according to the design.

Up to this point we have talked of specifications only at the informal level taking them to be representations with realizable content. More precisely, we take a specification to be a consistent set of properties aimed to constrain an activity or its outcome (e.g., to guide the realization of an envisioned product). Ontologically speaking, a specification is an object (essentially a set of properties) created by an agent. Any object that satisfies the constraints is a realization of the specification. Any consistent set of properties (where consistent means that an entity satisfying those properties does not contradict natural laws) can be a specification provided it is intentionally chosen (created) by an agent at some point in time. In other words, given a specification, there exists a set of its realizations (the set may be empty when no realization has been produced). Vice versa, given a finite (non-empty) set of entities, one can generate a specification (usually, more than one) that includes the entities. When all and only the members of that set are realizations of the specification, there exists a class whose extension is the set and intension is the specification. Here are some consequences of the distinctions we have introduced.

(1) Drama vs. Novel

Drama and *novel* are ontologically different from the representation viewpoint. A drama is a specification and is realized by actors that perform the sequence of activities described in the drama, while a novel does not have such performance. Of course, readers read the novel, but reading does not count as performance. Instead, reading of a novel is comparable to attending (as spectator) a drama performance, listening to a musical sound sequence or watching a painting (the object). All these activities are not performances, they are perceptions.

(2) Piece of music
A composer composes a specification of the sound sequence he/she creates. While there can be various performances of a piece of music, say, Beethoven's 5th Symphony (B5thS), and any two performances of that piece of music differ, all of them are understood as the B5thS because all the performances satisfy the rules coded in Beethoven's musical score (with some tolerance, of course). Given the set of all possible B5thS performances, Beethoven's piece of music is the commonality among all the performances, that is, a specification. What one listens to is a realization of the B5thS and it is an instance of musical sound.

(3) Procedure
The same logic applies to *procedure* which is a specification. What the creator of a procedure creates, say, Hoar's quicksort algorithm, is an instance of *procedure*. The procedure is written down in a form (an expression in a suitable language) and such form may be realized on a sheet of paper[4]. The actual form depends on the chosen language but the content of the representation is the Quicksort algorithm. All the actions performed by following the Quicksort algorithm are realizations of the specification of Quicksort algorithm. Sometimes people talk of the running of the algorithm as an instance of it but, from our discussion, this way of speaking is ontologically wrong. The running of the algorithm is a realization of the Quicksort algorithm and an instance of a sorting action.

(4) Car model
The same applies to a *car model* by which we mean the content of a representation. The Prius car model is a specification. Mrs. Alice's Prius is an instance of a car and is a realization of the specification, i.e, of the Prius car model. A catalog of car models is a representing thing which may contain several car models as representations. We discuss representing things in the next section.

4. Representing Thing

Consider a published book. When we buy it in a bookstore we can say that we bought the book but, when pressed about what we mean, we clarify that we got a *copy* of the book. This leaves open how to ontologically understand the object that the author wrote. To answer this question, we need to understand the distinction between *representation* and *representing thing*. Recall that a representation exists in time but is not a concrete entity. A representation is created by an agent but is not material. This allows us to say that the sentence "This is a door" denotes two entities: on the one hand a representation that has form (in English) composed of a sequence of 11 things denoting alphabet letters (independently of the font or handwriting, i.e., of the patterns of the letter), on the other hand what one understands when reading "This is a door". Therefore, expression "This is a door" as a representation (an individual) is not a physical object (it does not exist in space). One can print (by assigning letters and fonts to the form), say on a sheet of paper, a series of "ink marks", we call this sequence of ink marks the *representing thing* of representation "This is a door". In this way the representation (the form and content pair) is associated with a physical individual that we call a *representing thing*. A representing thing is a form-realization and has two parts: the representation medium, namely, the material support (the sheet of paper with ink marks in this case) and the representation itself.

---

[4] More precisely, this is a representing thing as we will see later.

The distinction between representation and representing thing enables us to distinguish the book that the author wrote, which is font-independent and thus a representation, from a representing thing, a copy of the book. The distinction between representation and representing thing is central and shows the importance of introducing non-physical entities in modeling representations. It also enables to make fine distinctions regarding the copies[5] of a representation. What is copied is what one can perceive on the representation medium (in the case of a physical book, the medium is the sheet of paper). Copying is just a generation of the 'same' (as detailed in Sect. 6.3) representation on a different medium. In fact, a novel, say, Tale of Genji, exists in the form of a book. A copy of a book is a 3D entity physically divided in distinct pages (a complex medium) with the form realized in chapters, sections, etc. The content of the copy of the book is the content of the representation of the representing thing and is associated with the ink marks that represent the form expressions in natural language and/or images. In this sense we can say that the Tale of Genji exists independently of the medium on which it is written. The separation between *representation* and *representing thing* will be deepened in the following sections.

5. A Formalization of the Ontology of Representations

5.1 Representation and Representing Thing

In this section we provide a minimal characterization of the theory of representation as introduced in this paper. The aim of this section is to highlight the basic elements and their relationships, in particular relatively to notions like realization, sentence and pattern. This logical theory is also introduced with a modular perspective: the axiomatization should be aligned with the (possibly foundational) ontology one is using and further specialized depending on the characteristics of the latter. Note that the module is not seen as suitable for arbitrary ontologies as the theory relies on specific choices. In particular, it assumes that along the traditional notion of object there are semi-abstract entities. This is the case, for instance, of Yamato with the categories Physical and Semi-abstract, and of Dolce with the categories Physical Object (POB) and Non-Physical Object (NPOB). An example of semi-abstract entity, as understood here, is *content.* It is a non-material entity and it starts to exist at some point in time. Content is a special type of semi-abstract since it can be encoded, for instance in a sentence. A form, as understood in this paper, is also a semi-abstract.

A typical, yet complex, example of representation is given by natural language sentences. The form of a sentence in natural language encodes the meaning of the sentence as usual. Consider now a single letter. The form of the letter is a linear drawing (think of it as a non-localized image as to distinguish it from a representing thing) which encodes the standard pattern of the letter as its content. For a different example, in a sculpture the form is the 3D shape (again, not localized) which encodes the content of the sculpture. These forms are intrinsically 'spatial' because, even if not localized, they require spatial dimensions to exist. To understand these intrinsically spatial forms as semi-abstract, one can think of a digital representation in which a 2D or 3D image in the real world is captured in a digital form. This form can be realized on a display for human perception. In ontological terms, such digital representation is a specification of the 2D (or 3D) pattern. This specification is the form, and it is neither the content of the sentence

---

[5] This means not only photo-copies but also transcriptions.

nor that of the statue. This *form-specification* is also not to be confused with the specification of, say, a procedure, which is the content of the procedure.

Coming back to individual letters (characters), the form of a letter as a specification is about the ***form-realization*** (the localizing of a letter pattern), while the content as a specification is about the ***content-realization*** (the drawing of a letter). Thus, in the case of letters the form-specification is a specification of the particular pattern of the letter to be localized, and that content-specification is a specification of the standard pattern of the letter as a specific element of the letter alphabet (in general a letter may be associated with more than one pattern, for simplicity here we assume there is only one).

First, we define representations as entities composed of two parts, a form and a content, connected by an encoding relationship with its related encoding method

$$(Df1)\ \ representation(x,y,z,m) =_{def} form(y) \wedge content(z) \wedge method(m) \\ \wedge encodes\_via(y,z,m) \wedge x = y + z$$

The encoding relationship that we consider applies to form, content and method (in this order) and is functional

$$(Ax1)\ \ encodes\_via(x,y,z) \rightarrow form(x) \wedge content(y) \wedge method(z)$$
$$(Ax2)\ \ encodes\_via(x,y,z) \wedge encodes\_via(x,y',z) \rightarrow y = y'$$
$$(Ax3)\ \ encodes\_via(x,y,z) \wedge encodes\_via(x',y,z) \rightarrow x = x'$$
$$(Ax4)\ \ encodes\_via(x,y,z) \wedge encodes\_via(x,y,z') \rightarrow z = z'$$

Axioms (2)-(4) show that methods, as understood here, are very detailed, they imply that two distinct forms with identical content (as in the case of a translation) must have been obtained via different methods. Also, methods can be combined: if *m* and *m'* are methods, then their functional composition, *m'(m(x))*, is also a method provided the *m(x)* falls within the domain of *m'*.

It is helpful to define a representation predicate as follows

$$(Df2)\ \ representation(x) =_{def} \exists y,z,m\ representation(x,y,z,m)$$

It follows that the encoding relationship characterizes representations

$$(Th1)\ \ encodes\_via(x,y,m) \wedge v = x + y \rightarrow representation(v)$$

Whenever a representation exists, so does its form and content and so does the encoding method that binds them

$$(Ax5)\ \ representation(x,y,z,m) \wedge Pre(x,t) \rightarrow Pre(y,t) \wedge Pre(z,t) \wedge Pre(m,t)$$

A few observations are needed. First, we have not temporally characterized the encoding. This is a formal (atemporal) relationship. However, whenever the representation is present, so does the form, the content and the encoding. While the encoding does not depend on time, a form and a content are such in so far as they satisfy a suitable encoding relationship. Second, it is not possible for a form to exist without some content associated with it, and vice versa. Third, the same form (content) can be associated with different contents (forms) provided each encoding is done with a different method. Finally, we do not make any commitment on the possibility of form, content and method to disappear, i.e., if they cease to exist at some point or for some reason. The theory is neutral on these aspects.

Representation, form and content exist in time, but neither is located in space because they are semi-abstract.

$$(Ax6)\ \ representation(x) \lor form(x) \lor content(x) \\ \rightarrow \exists t\ Pre(x,t) \land \forall y,t'\ \neg Located(x,y,t')$$

Given a representation x, for simplicity and when there is no risk of confusion, we write x_f for the representation form and x_c for the representation content. A representing thing is the sum of a representation and a representation medium (the material support) which is a realization of the representation form

$$(Df3)\ \ representing\_thing(rt,x,y) =_{def} representation\_medium(x) \\ \land representation(y) \land rt = x + y \land form\_realization\left(x, y_f\right)$$

The entity representing thing is problematic in several foundational ontologies since it is the sum of a physical and a non-physical entity. In Dolce it is classified as Arbitrary Sum, in Yamato as Independent Entity. In an ontology that does not allow this kind of entities, one could model them indirectly as dependences across physical and semi-abstract entities.

$$(Ax7)\ \ representation\_medium(x) \rightarrow physical\_object(x)$$
$$(Ax8)\ \ form\_realization(x,y) \\ \rightarrow representation\_medium(x) \land form(y) \\ \land \exists rvm\ (representation(r,y,v,m) \land Part(r,x))$$
$$(Ax9)\ \ form\_realization(x,y) \land representation(r,y,u,m) \land Pre(x,t) \\ \rightarrow Pre(r,t)$$

Then, we define the realization of content, as in the case of a procedure, as follows (here we use the standard satisfaction relation $satisfies(x,y)$ to mean that the entity x satisfies the specification y).

$$(Df4)\ content\_realization(u,v) =_{def} (physical\_object(u) \lor event(u)) \\ \land content(v) \land satisfies(u,v)$$

We now concentrate on the case of sentences in natural language. As we have seen in the previous section, this case is particularly rich and complex to model. Here we aim to give a formal description of the basic elements of the theory. To model the realization of a form (form-realization) we need to refer to: representation, ink-mark, ink-pattern and pattern. The pattern is what a writer chooses when producing the representing-thing, say a manuscript or a printed book, and it corresponds to the font used in printing. (Here we do not attempt to characterize the pattern itself.) There are a few relationships that coordinate the dependences between representation and representing thing in (written) expressions of natural language: ink_pattern and pattern_of. We write $ink\_pattern(x,y)$ to mean that x is a pattern to be realized by a marking substance like ink (yet, in practice it can be electronic) of a representation form y. The pattern itself is semi-abstract. We write $pattern\_of(x,z)$ to mean that x is the pattern realized by the mark z (of ink, in this case).

$$(Ax10)\ \ ink\_pattern(x,y) \rightarrow pattern(x) \land form(y)$$
$$(Ax11)\ \ pattern(x) \rightarrow \exists t\ Pre(x,t) \land \forall y,t'\ \neg Located(x,y,t')$$
$$(Ax12)\ \ pattern\_of(x,y) \rightarrow pattern(x) \land ink\_mark(y)$$
$$(Ax13)\ \ ink\_mark(x) \rightarrow physical\_object(x)$$

Given these relationships, we can define the realization of a form in natural language ($NLform\_realization$) which must constrain the relationships between the form, the pattern and the mark.

$$(Df5)\ NLform\_realization(u,v) =_{def} form\_realization(u,v) \\ \wedge \exists p,m\ ink\_pattern(p,v) \wedge ink\_mark(m) \wedge pattern\_of(p,m) \\ \wedge part(m,u)$$

Remaining on the case of natural language, one could go further and define a sequence of symbols (letters of the alphabet) associated with a form. From this, it is possible to give the notion of sentence as a sequence of symbols that satisfies the language specification (the grammar), call it $sentence\_as\_form$. Then, we can define the representation of a sentence with this formula

$$(Df6)\ sentence\_representation(r) =_{def} representation(r) \\ \wedge sentence\_as\_form(r_f) \wedge \exists z\ method(z) \wedge encodes\_via(r_f, r_c, z)$$

To turn a symbolic representation into a representing thing, which we remind the readers counts as a form-realization rather than a content-realization, one first fixes the font and its size, so that the ink patterns are determined (these are realized as ink marks on a sheet of paper when printed). A symbolic representation is thus a sequence of letters' identifiers which are not physical entities. However, in practice a representation is created directly as a representing thing. The need of finer distinctions as introduced here raises when one aims to understand the phenomena from the ontological viewpoint.

5.2 On Letters

Here we apply the theory to clarify the nature of letters (characters) seen as representations. A letter "a" written or printed on a sheet of paper is a tangible 2D object, a linear drawing which people recognize because the linear drawing encodes the commonly shared pattern "a". Leaving aside the specificity of the pattern and how it may evolve in time, we aim to uncover its contribution to the ontology of representation.

A letter's form encodes a pattern which can be the standard pattern of the letter, say, the pattern of letter "a". What is encoded in the letter form in addition to the standard pattern can be something else, namely, the letter as the first element of the alphabet. The essence of the pattern "a" is that it denotes the first letter of the alphabet. These two things should not be confused. Imagine replacing 'x1' for 'a', 'x2' for 'b', and so on. While computers can promptly process the new obtained sentences, this can be hard for humans. This failure (or serious difficulty) is caused by the lack of an encoded association between the new pattern and the identity of the letter of the alphabet as the *content* of the representation of "a". The association between 'x1' and the alphabet letter 'a' must be recreated with practice. Of course, the letter as a representation object is not influenced by the replacement and the consequent change in the association.

To conclude, all the many printed letters, say, "a", "a" and "A" that we perceive in daily life are realizations rather than instances of letter "a". The linear drawing of a letter realizes a pattern of the letter. The realization is a form-realization. At the same time, it realizes the essential pattern of the letter since we do recognize the letter rather than a mere drawing. Thus, this is a content-realization. These observations show that when facing printed letters we are dealing with two distinct specifications. The former specification is the pattern of the written letter, the latter one is the standard (essential) pattern of the letter as an element of the alphabet. These two realizations produce one single (material) linear drawing.

5.3 Formalization of Letter as Representation

Now we can further specialize our previous theory relatively to expressions using letters. First, let us write $classifies(x, y)$ for the standard relationship x classifies y; $specifies(x, p)$ for x is a specification of pattern y; $letter\_pattern(x)$ for the predicate that holds for the standard pattern of a letter; $linear\_drawing(x)$ for the predicate that holds for the pattern that is realized; and $id\_alphabet(n, x)$ for the relationship that identifies the pattern of the n-th letter of the alphabet. We write $encodes\_letter(x, y, z)$ to mean an encoding (z) occurring in letter representations in which the content (y) classifies the form (x).

$$(Ax14)\ \ encodes\_letter(x, y, z) \rightarrow classifies(y, x) \wedge encodes\_via(x, y, z)$$

We can then say that the form of a letter is a specification of a pattern and that the content of a letter is either the letter of the alphabet or the standard pattern of the letter (here n is an integer ranging from 1 to the number of letters in the alphabet, which is a finite number).

$$(Ax15)\ \ letter\_form(x) \rightarrow \exists p\ specifies(x, p) \wedge linear\_drawing(p) \wedge form(x)$$
$$(Df7)\ \ letter\_content(x) =_{def} content(x) \wedge \exists n, p\ (id\_alphabet(n, p) \vee (specifies(x, p) \wedge letter\_pattern(p)))$$

Finally, we can define a letter representation

$$(Df8)\ \ letter\_representation(r, x, y, z) =_{def} representation(r, x, y, z) \wedge letter\_form(x) \wedge letter\_content(y) \wedge encodes\_letter(x, y, z)$$

An ontological difficulty that led to this theory is due to contrasting properties of letters: letters are at the same time symbolic and analogue entities. By distinguishing the symbolic aspect (the element in the alphabet) from the analogue aspect (the letter pattern), we obtain a coherent representation theory which can clarify their relationship. Accordingly, letters as form of a symbolic representation are dealt with as identifiers of entities of the alphabet, letters in the form of representing thing are dealt with as analogue representations whose form is a linear drawing. As the formalization shows, what appears in the form of a symbolic representation is the letter as the content, i.e., the identifier of the letter in the alphabet.

6. Discussion

6.1 Classifying Representation Entities

In Table 1 we summarize the entities we have discussed addressing the ontology of representation. *Novel* has no direct physical entity in the real world. There are corresponding physical entities that are realizations like books, eBooks etc., that is, representing things. The other representations are associated with two types of physical entities. This difference comes from the fact that a representation which has a *designed content* has two ways of realization: one realization is a product according to the specification (content-realization) and the other is a product according to the form (form-realization).

Table 1 Summary of representational entities.

	Representation	Content	Representing thing (form-realization)	Individual as content-realization
Industrial Product	Its design drawing/ specification	The specification of the structure/functions	The printed design drawing/specification	Itself
Painting	The digitized image of the painting displayed on a screen	What is painted such as persons, scenery...	The painting on a canvas	None
Music	The musical score	The specification of the sound sequence	The printed musical score	The sound sequence
Procedure	The description of the procedure	The specification of the actions/operations	The printed description of the procedure	The execution process of procedure
Letter	The letter	The specification of its standard pattern	The letter written on a sheet of paper	The linear drawing
Novel/book	The text (the novel one writes)	The meaning of the text	A copy of the book	None

Table 1 tells us that the representing thing (the form-realization) of a painting is the "painting on a canvas", what we usually see in a museum or art exposition. The representation that corresponds to it must be a specification of the form encoding the painting content. Like in the case of letter, the form of a painting is the specification of the "ink pattern", and the representation of a painting is the digitized image of the painting displayed on a screen. In this sense, *painting* is special: it is primarily what it shows, i.e., the form. In this it is like *handwriting in calligraphy*, essentially the form that it manifests. On the contrary, *novel* has both a visible aspect (ink-patterns/marks of letters and sentences) and a non-visible aspect, its content (the story).

As we already discussed, entities obtained by the form-realization and the content-realization of a letter coincide with each other in reality. In the table, we distinguish them by using different expressions to reflect the differences of the realization processes: "letter written on a sheet of paper" for the former and "linear drawing" for the latter.

6.2 On (the Missing) Content

Precisely speaking, sentences have two *contents*: the meaning of the sentence and the style (beauty) of the sentence. The discussion thus far only deals with the former content. To make the discussion more comprehensive, the latter content needs to be incorporated in the theory. A good writer has his/her own style of sentences which is the heart of writing. In this view, a *natural language expression* which is classified into the *representation form* has also another *content*, we may call it *style-content*, which is the specification of his/her *sentence style.* This style-content requires its specific relationships but does not need further discussions as the formalization is essentially similar by the machinery already developed for the meaning of a sentence (see also [3]).

6.3 Copying and Identity

Given a representation whose form is a sequence of symbols, one can copy the sequence of symbols or copy the sequence of the symbols' images (photocopying). In both cases, what one copies is the representing thing. Since symbols are unique, one cannot really copy just symbols: one copies a representing thing which is a realization of symbols. Copying is thus the realization of the *same representation* on a different medium. The issue is what "same" means. Roughly speaking, here two kinds of sameness are present.

One is given by a representing thing that realizes the representation form of another representing thing via different fonts or handwritings. Here sameness means that it is a realization of the same form (thus, the very same symbols). The other kind of sameness corresponds to an act of photocopying, that is, to obtain a representing thing that realizes the representation form of another representing thing via the same ink patterns (the ink marks of the two representing things have the same pattern).

The distinction between copying and photocopying tells us about relative identity of representations and of representing things. There are four levels of identity to keep in mind: (0) content level, (1) symbol level, (2) ink pattern level and (3) medium level. A perfect translation using different languages, say from a logical theory to an equivalent theory formulated in a different logical language, is a copy at the level (0). These entities are identical with respect to content. A hand-written copy of a sentence on a sheet of paper is identical to the original one at the level (0) and (1), as they realize the same symbols. A photocopy of a sentence on a sheet of paper is identical to the original one at the level (0), (1) and (2), as they realize the same patterns (modulo the quality of the photocopy). At the medium level, two physical entities cannot be identical as we assume that the medium in the two representation things must be different.

7. Related Work

A comprehensive state of the art on informational objects has been presented recently [12]. Philosophy of language and thought [14] (and so approaches in natural language understanding and processing), treats syntactic expressions (the form in our term) and meaning (the content in our term) as different entities to separate syntactic and semantic analyses as far as possible. In our ontology of representation, the form and the content constitute a single entity, called representation, due to an encoding relationship. This is mainly because the aim of our theory is to provide an ontological understanding of a "common-sense view of the world" rather than tools for natural language processing/ understanding.

The ontology of literary works discussed in philosophy by Thomasson [13] takes the view that a particular sound sequence is not an instance of a class. Our theory follows this intuition and adds that this sequence should be understood as a realization of a piece of music. Thomasson discusses neither specification nor our development of the symbols/ letters relationships. Also, conventional ontologies of literary works [12] [13] concentrate on how readers (users in our term) interpret/understand the works. Our ontology does not since we concentrate on foundational aspects which are domain independent. We support the idea that, ontologically, any representation including literary works exists independently of its readers/users.

One of the unsolved issues in ontology of literary works is whether all literary works share the same ontological status [13]. In fact, people tend to believe that novels are abstract objects and paintings are concrete objects. Our ontology gives a comprehensive view to this issue by providing a notion, that of representation, that subsumes *novels, poems, paintings,* and *sculptures*.

Since this paper aims to develop a refined theory of information objects as discussed in applied ontology, perhaps the main theory to discuss is the Information Artifact Ontology (IAO) approach presented in [1][2]. Even though strictly related, IAO and our theory focus on different entities. The former discusses information, the latter centers on representations and takes information to be a role played by a representation under the context of an informing activity.

The term Information Content Entity (ICE), the main type modeled in IAO, turns out to be ambiguous when analyzed from our theory's viewpoint. A terminology match can help to clarify this. Content in ICE combines both *content* and *form* in our terminology. IQE, which is a concretization of ICE, corresponds to *representing thing* in our theory. Although some IQEs are said to be concretization of ICE, examples of ICE are not given except in the following excerpt from [2]:

> *...for example, the dependent entity which is the pattern of ink marks in your copy of the novel War and Peace (a complex quality in BFO terms) – are able to migrate from one bearer to another (e.g., through use of a photocopier).*

This piece suggests that the dependent entity made of the ink patterns is an ICE because it satisfies the definition of ICE. This does not seem to follow from the guidelines of the IAO approach. An ICE is what is ready to be concretized in a *representing thing* (in our terminology) and has strong association with how to achieve the concretization as seen in *"this sentence is concretized in this pattern of ink marks on this piece of paper"* [2]. That is, the definition of ICE covers not only *content* but also *content* encoded in *form* (the above example). Hence the shapes/patterns of letters seem to be embedded in sentences. In our theory of representation, on the other hand, the separation is clearly made as the duality of letters is explicitly modeled: letters appear as identifiers in sentences and are analogue representations in themselves. A sentence in our theory, in the sense of *representation form*, is composed not of specific letters of specified fonts which explicitly have their own "ink pattern" but of identifiers of letters independently of the "ink pattern". In short, a sentence in IAO is ink-pattern-dependent, while in our theory it is not. This might be mainly due to the so-called realist approach [9] of the underlying ontology and the emphasis on *representing thing-centered* entities.

8. Concluding Remarks

Representation, in our view, is more fundamental than information and more approachable for ontological analysis. Because of this, here we studied ontological representation objects (ORO), their ontological nature and an initial axiomatization. There are three types of content: denotation, specification and description. We focused on specification which characterizes representation where there are performers as content-realizers, as well as creators, like in procedures and pieces of music. A clear separation between representation (ORO) and representing thing enabled us to identify the ontological status of a book and of what one writes beyond the common yet naïve separation between content and material support. The fact that what one writes is font-independent shows that a book one writes is an ORO rather than a representing thing.

This paper presents only the basics of an ontology of representation and much is left unexplored. They plan to expand this work towards a unified theory of representation that includes a full explanation of how to model letters, contribute to move towards a more solid ontological foundation of information objects like data, information and knowledge.

References

[1] Ceusters W. An information artifact ontology perspective on data collections and associated representational artifacts. Stud Health, Technol Inform. 2012; 180: 68-72.

[2] Ceusters W, Smith B. Aboutness: Towards Foundations for the Information Artifact Ontology. Proc. of the Sixth International Conference on Biomedical Ontology (ICBO); 2015 (CEUR 1515).

[3] Mizoguchi R. Advanced course of ontological engineering Part 3, Section 2 Ontology of representation. New Generation Computing; 2004; 22 (2): 193-220.
[4] Mizoguchi R. YAMATO: yet another more advanced top-level ontology. Proceedings of the Sixth Australasian Ontology Workshop; 2010: 1-16.
[5] Yablo S. Aboutness. Princeton University Press. 2014.
[6] Zins C. Conceptual approaches for defining data, information, and knowledge. Journal of the American society for information science and technology. 2007; 58(4): 479-493.
[7] Lewis D. Convention: A philosophical study. John Wiley & Sons; 2008.
[8] Ni P. Moral and philosophical implications of Chinese calligraphy. Grand Valley Review. 1999; 20.1.
[9] Smith B, Ceusters W. Ontological realism: A methodology for coordinated evolution of scientific ontologies, Applied Ontology. 2010; 5 (3):139-188.
[10] Mizoguchi R, Sunagawa E, Kozaki K, Kitamura Y. The model of roles within an ontology development tool: Hozo, Applied Ontology. 2007; 2 (2), 159-179.
[11] Mizoguchi R, Galton A, Kitamura Y, Kozaki K. Families of roles: A new theory of occurrent-dependent roles, Applied Ontology. 2015; 10 (3-4), 367-399.
[12] Sanfilippo EM. Ontologies for information entities: State of the art and open challenges. Applied ontology. 2021; 16 (2):111-135.
[13] Thomasson AL. The ontology of literary works. In The Routledge Companion to Philosophy of Literature. 2016; Routledge; 349–358.
[14] Dummett M. Frege: Philosophy of Language. Harvard University Press. 2nd edition. 1981.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210371

Towards a Unified Dispositional Framework for Realizable Entities

Fumiaki TOYOSHIMA[a,1], Adrien BARTON[b,a], Ludger JANSEN[c]
and Jean-François ETHIER[a]
[a] *GRIIS, Université de Sherbrooke, Quebec, Canada*
[b] *Institut de Recherche en Informatique de Toulouse, CNRS, Toulouse, France*
[c] *Institute for Philosophy, University of Rostock, Germany*

Abstract. Realizable entities are properties that can be realized in processes of specific correlated types in which the bearer participates. It will be valuable to create a systematic classification of realizable entities because they are useful for various modeling purposes in ontologies. In this paper we outline a unifying framework for realizable entities (including dispositions and roles) in the upper ontology Basic Formal Ontology (BFO) that is theoretically underpinned by J. McKitrick's pragmatic approach to dispositions. In particular, we develop a formal ontological account of "extrinsic dispositions" and illustrate its potential applications with clarification of functions and roles in BFO.

Keywords. realizable, disposition, extrinsic disposition, role, Basic Formal Ontology (BFO)

1. Introduction

The world is teeming with realizable entities: properties that can be realized in processes of specific correlated types in which the bearer (e.g., material objects) participates. For example, the fragility of this glass can be realized in a process of the glass breaking. Realizable entities and the realization relation between properties and processes are vital for considering the interplay between two major upper-level ontological categories: continuants (aka endurants), in particular material objects, and occurrents (aka perdurants), in particular processes. For instance, Guarino & Guizzardi [1] propose the view of events as *manifestations* (realizations) of "individual qualities" (property particulars) and Guarino [2] expects that this manifestation account of events will inspire many upper ontologies. More specifically, dispositions (e.g., fragility) among other things have been intensively investigated in formal ontology [3][4][5] and they have been deployed in the building of many domain ontologies (see Toyoshima's [6] general survey). The disposition category is adopted by some upper ontologies that have core features (e.g., the continuant/occurrent distinction) in common, such as Basic Formal Ontology (BFO) [7][8] and the Unified Foundational Ontology (UFO) [9][10]. Accordingly, realizable entities are useful for modeling a wide range of domain-specific entities, such as suicidal tendencies in medical informatics [11]. It nonetheless remains

---

[1] FT acknowledges financial support by the SPOR Canadian Data Platform (CIHR).

largely unexplored how to articulate different realizable entities *systematically*, notwithstanding the line of study started by Röhl & Jansen [12].

In this paper, we will embark upon a systematic study of realizable entities. For this purpose, we examine the category of realizable entity in BFO, especially its prominent direct subtypes: dispositions and roles. As for dispositions, there is long-standing controversy as to their ontological nature. For example, Guarino [2] says: "the difficulty of distinguishing one disposition from another (…) is a good evidence of their problematic ontological status" (p. 14). Barton et al. [5] propose a set of identity criteria for dispositions that can meet this challenge, but a more fundamental question is what it is like for a property to be a disposition. Consider the following alleged "recipe" for identifying dispositions [13]: one can identify a disposition by using an expression of the form "the disposition to bring about *R* if *T* holds" where *R* is a realization (e.g., the glass-breaking) and *T* is a triggering condition (e.g., the glass being forcefully pressed). From an ontological point of view, however, the reliability of this recipe remains unclear. For instance, the recipe could yield the "disposition to get harmed if attacked", but am I really *disposed* to get harmed if attacked? If we are agnostic as to such dispositions, how should we treat this recipe-based identification of dispositions in formal ontology?

The issue of dispositions is further complicated by the BFO characterization of roles as a disjoint subtype of realizable entities. For one thing, this realizable understanding of roles may be sometimes regarded as contentious. Guarino [2] states that it: "reflects a very peculiar understanding of the role notion which, although useful, would require a broader framework" (p. 14) and one reason why he thinks so is the difficulty of figuring out the relationship between dispositions and roles in BFO. (We will discuss his concern in more detail later.) Assuming that being a student consists in studying at school, for instance, exactly how is Mary's role of being a student at school linked with her disposition to study? To borrow Guarino's expression, addressing these questions requires careful consideration of dispositions and roles within "a broader framework", presumably within a systematic framework for realizable entities in general.

The paper explores a systematic perspective on realizable entities by revolving around the BFO framework. Section 2 describes how BFO represents realizable entities, dispositions, and roles. Section 3 develops one way of classifying systematically realizable entities, taking a cue from McKitrick's [14] pragmatically motivated approach to dispositions. Section 4 is devoted to the discussion on the potential application of our proposal. Section 5 concludes the paper. In the text, we will write terms for type-level entities (aka universals, or classes) in italics and terms for token-level entities (aka particulars, instances) and relations in bold, respectively.

In formalization, variables stand only for tokens, predicates (written in bold) stand for types (unary predicates) and relations, and free variables are universally quantified. We will employ conventional logical symbols of first-order logic, including "¬", "→", and "∃". Table 1 lists relational predicates together with their informal explanation and most of them represent binary relations, which can have the practical virtue of enhancing the practical implementation of our proposal in information systems, such as the ones constructed in the Web Ontology Language (OWL).

2. Realizable Entities in Basic Formal Ontology

We begin by adumbrating the basic background of BFO before explaining its category of realizable entity. BFO is an upper ontology that is rooted in the realist methodology

Table 1. A list of relational predicates and their informal explanation. (The references therein mean prior work from which relations are imported, although they may be reinterpreted in our context.)

Relational predicate	Informal explanation
ADP(x,y) [4]	x (realizable entity) is an add-part of y (realizable entity)
HCB(x,y) [3][5]	x (realizable entity) has as categorical basis y (quality)
HTR(x,y) [3][5]	x (realizable entity) has as trigger y (process)
INH(x,y) [7][8]	x (specifically dependent continuant) inheres in y (independent continuant)
REAL(x,y) [3][5]	x (realizable entity) can be realized in y (process)
REON(x,y)	x (extrinsic disposition) relies existentially on y (intrinsic disposition)
SUM(x, y_1, ..., y_{n+1})	x is a mereological sum of y_1, …, y_{n+1}

for ontology development [15]: ontologies should represent entities in actual (scientific) reality. It has the top-level distinction between continuants and occurrents, the former being further divided into independent continuants and dependent continuants. Among dependent continuants are specifically dependent continuants, which depend (existentially) on at least one independent continuant.[2] As for occurrents, we will focus on one of its subcategories, namely processes: occurrents that exist in time by occurring, have temporal parts and depend on at least one independent continuant as participant. Two major subtypes of specifically dependent continuants are realizable entities (which will be detailed below) and qualities: specifically dependent continuants (e.g., color, shape, and mass) that do not require any further process in order to be realized.

A realizable entity is a specifically dependent continuant that inheres in some independent continuant and is of a type some instances of which are realized in processes of a correlated type. BFO identifies two immediate subtypes of realizable entities, namely dispositions and roles. First of all, a role in BFO is: "a realizable entity that (1) exists because the bearer is in some special physical, social, or institutional set of circumstances in which the bearer does not have to be (optionality), and (2) is not such that, if this realizable entity ceases to exist, then the physical make-up of the bearer is thereby changed (external grounding)" ([7], pp. 99-100). Therefore, a role is an optional and externally grounded realizable entity. Suppose for instance that Mary is a student at the XYZ college. Mary has the role of being a student (which may be realized in, e.g., a process of Mary's studying) because she happens to be in some specific institutional circumstances with respect to the XYZ college (optionality) and she does not undergo physical changes just because she ceases to be a student (externally grounded).

By contrast, a disposition in BFO is: "A realizable entity (…) that exists because of certain features of the physical makeup [material basis] of the independent continuant that is its bearer" ([7], p. 178). BFO also describes a disposition as an *internally grounded* realizable entity: if a disposition ceases to exist, then the physical makeup of the bearer is thereby changed. To use a canonical example, the fragility of this glass can be realized in a process (realization) of breaking when it is pressed with force, it is based on some structured molecules (material basis) of the glass, and the glass is physically changed when it is no longer fragile (internally grounded).

Thus, BFO characterizes dispositions as internally grounded realizable entities. BFO employs what may be called the "grounded test" for realizable entities. That is to say, a realizable entity is internally grounded if and only if its bearer needs to be physically changed for it to cease to exist, and it is externally grounded if the realizable can cease

---

[2] It is sometimes said that processes can be bearers of properties, such processes of oscillation having waves as property [16]. BFO seems to refrain from this possibility, but we may think that the bearer view of processes could be an auxiliary assumption that would simplify our formalization below.

to exist without its bearer changing physically. Consequently, a realizable entity is a disposition iff it passes the internally grounded test.

In contrast, roles are not only externally grounded but also "optional". The notion of optionality used there may be less clear to comprehend than grounded-ness. As Röhl & Jansen [12] say, one way to clarify it would be to utilize Guarino & Welty's [17] work on the classification of types based on their notion of rigidity. They define rigidity, semi-rigidity, anti-rigidity, and non-rigidity of types as follows:

- A type is *rigid* if it is essential to all its possible instances.
- A type is *semi-rigid* if it is essential to some of its possible instances and not essential to others.
- A type is *anti-rigid* if it is non-essential to all its possible instances.
- A type is *non-rigid* iff it is either semi-rigid or anti-rigid.

We can elucidate optionality (and "non-optionality") in terms of these features, while prescinding from exactly how the notion of rigidity can be integrated into the BFO framework (see Seyed & Shapiro's [18] inquiry into this integration). Assuming that roles are type-level entities, Guarino & Welty [17] think that role types are anti-rigid because: "Roles are properties that characterize the way something participates [in] a *contingent* event or state of affairs" (p. 214). Every instance (e.g., Mary) of *Student* can cease to be such in a suitable situation (e.g., her graduation from the XYZ college), for example. One way to elucidate optionality is thus to construe it as anti-rigidity, and non-optionality as either rigid or semi-rigid. Another way is to understand optionality and non-optionality as non-rigidity and rigidity, respectively, *à la* Röhl & Jansen [12], who highlight a category of externally grounded and non-optional realizable entities that is not presently acknowledged by BFO. In short, rigid and anti-rigid types of externally grounded realizable entities would be safely taken to be non-optional and optional, respectively; whereas, the treatment of their semi-rigid types may be an open question.

Two clarificatory remarks on roles in BFO are in order here. First, the phrase "play a role" has been popular in formal ontology and other domains, but roles in BFO are something to be *had* or *borne*. To see this point, it will be useful to mention Mizoguchi et al.'s [19] claim that the alleged "role-playing" has two components: to *hold* a role and to *perform* a role. To take a variant of their example, Mary is still a student of the XYZ college when she is asleep at home because sleeping Mary can hold a student role (because she has enrolled in this college), but she does not perform it: that is, she does not do anything associated with her being a student. This distinction between holding and performing a role corresponds to having a role and this role being realized within the BFO framework, respectively: when asleep, Mary has a role of being a student and this role remains unrealized.

Second, Guarino [2] says that "BFO can only account for a notion that is *related* to the ordinary notion of social role" but not "a social role in the ordinary sense" (p. 14). According to Toyoshima [20], this may be partly due to multiple possible understandings of role terms. Consider the statement: "Mary is a student." For instance, UFO [9][10] takes *Role* to be a subtype of what UFO calls "anti-rigid sortal universal" and Mary to be an instance of *Student* which is a subtype of UFO:*Role*. In this account, one can identify the referent of the English term "student" with an instance of *Student*. On the BFO account, by contrast, a student like Mary is not an instance of *Student role* (which is a subtype of BFO:*Role*), since Mary is not a realizable entity. Rather, Mary *has* (rather than *being*) an instance of BFO:*Role* of being a student. Moreover, the term "student" can be generally defined as a person who has some role of student ([7], p. 100).

3. Towards Systemization of Realizable Entities

3.1. J. McKitrick's Dispositional Pluralism

By and large, philosophical discussions of dispositions focus primarily on physical properties (e.g., electric charges) to articulate the fundamental fabric of the world. In formal ontology, by contrast, domain experts are interested in dispositions as a convenient conceptual tool for representing a broad range of entities, as is well illustrated by the dispositional theory of diseases that is given by the BFO-compliant Ontology for General Medical Science (OGMS) [21] and a dispositional approach to such mental entities as belief, desire, and intention [22]. In other words, formal ontologists need an account of dispositions that is wide enough to characterize multifarious entities.

This practical attitude towards dispositions can be theoretically undergirded by McKitrick's [14] theory of dispositions that is pragmatically motivated by a dispositional analysis of various entities such as character traits and gender [23] (refer to Toyoshima et al. [24] for a formal analysis of her dispositional theory of gender). She argues for what she calls "five marks of dispositionality", according to which a property (instance) is a disposition if it ([14], p. 2):

1. has some characteristic manifestation ["realization" in our terminology] M [type];
2. is such that some circumstance C [type] will trigger manifestation M;
3. can be possessed without manifestation M occurring;
4. is instantiated ["borne"] by things [bearers] of which a conditional of the form "if it were subject to circumstance C, it would exhibit manifestation M" is generally true; and
5. can be accurately characterized with an expression of the form "the disposition to produce manifestation M in circumstance C."

Notice that her five marks can be seen as a more sophisticated version of the "recipe" for identifying dispositions that was alluded to in Section 1.

Based on these five marks, McKitrick endorses what she calls "dispositional pluralism", i.e., the thesis that dispositions are an abundant and diverse group of properties. For instance, it is prevailing orthodoxy that dispositions are intrinsic properties. The distinction between intrinsic and extrinsic properties is notoriously difficult to be defined explicitly, but the basic idea is that a property instance is *intrinsic* if it inheres in its bearer purely in virtue of the way its bearer is and it is *extrinsic* otherwise [25]. The fragility of a certain glass is intrinsic because the glass is fragile under any external circumstances, even when packed in a bubble wrap. In her pluralistic approach, however, McKitrick argues that there are also "extrinsic dispositions" (see Section 3.3), as well as many other non-ordinary kinds of dispositions, such as "ungrounded dispositions" (see Section 3.2).

We hypothesize that McKitrick's dispositional pluralism can be leveraged as one promising systematic approach to realizable entities, including BFO:Disposition. For one thing, this doctrine goes far beyond the BFO category of disposition: for example, McKitrick's extrinsic dispositions are externally grounded: an extrinsic disposition is borne (at least partly) in connection with the world that is external to its bearer, and when the external world changes, it can cease to exist even without the bearer's physical changes, hence the failure to pass the internally grounded test. (Relatedly, intrinsic dispositions are internally grounded.)

Dispositional pluralism can be expected to provide, so to speak, a "dispositional lens" through which to view realizable entities consistently. This dispositional lens will help to analyze more meticulously, for instance, roles and their relationship with dispositions in BFO. To center around dispositions in order to explore realizable entities in general will also enable taking advantage of the fact (mentioned in Section 1) that dispositions are the best-investigated type of realizable entities in formal ontology.

As a preliminary to formalization, we provide the following *is-a* hierarchy of mutually exclusive ontological categories with their corresponding unary predicates (where a type *A* being a subtype of a type *B* implies all instances of *A* being instances of *B*):

- Realizable entity (REA)
  - Intrinsic realizable entity
    - Intrinsic disposition (IND)
      - Ungrounded disposition (UND)
  - Extrinsic realizable entity
    - Extrinsic disposition (EXD)

Note that realizable entities in our class hierarchy can be identified with realizable entities in the BFO framework. Our classification of realizable entities will thus serve to characterize subtypes of realizable entities in BFO. For instance, intrinsic and extrinsic dispositions (which will be closely scrutinized in Section 3.3) can be located within the BFO hierarchy as follows:

- BFO:Specifically dependent continuant
  - BFO:Quality
  - BFO:Realizable entity
    - BFO:Disposition ("internally grounded realizable entity")
      - Intrinsic disposition
    - Extrinsic disposition

3.2. The Causal Import of Realizable Entities

We begin with the basic problem of what it means for an entity to be realizable. Assuming that dispositional pluralism is general enough to cover a large number of realizable entities, this would amount to the question of what dispositions in dispositional pluralism can have in common. To address it, we will consider McKitrick's [14] discussion on *causal bases* of dispositions. This notion should not be conflated with material bases of dispositions in BFO (see Section 2), not least because causal bases are properties of the disposition bearer but material bases are its material parts.

The pivotal idea is that the realizability of realizable entities would be construed in terms of their causal relevance to their realizations. For example, what makes the fragility of this glass realizable can be ascribed to an intimate causal connection between this fragility and a process of glass-breaking (in which it can be realized). In discussing the causal import of dispositions in her pluralist theory, McKitrick examines Prior et al.'s [26] two theses about causal bases of dispositions. We formulate them as follows:

- *The causal thesis*: Every disposition has a causal basis.
- *The distinctness thesis*: Every disposition has a distinct causal basis from itself.

McKitrick affirms the causal thesis but denies the distinctness thesis. In her understanding: “A causal basis is a property of an object which is, or would be, causally relevant to the manifestation [realization] of a disposition of that object” ([14], p. 132). Note that a disposition can have multiple causal bases because: “if some property is causally relevant to a manifestation [realization], this does not rule out some other property also being causally relevant to that manifestation. So we should not assume that there is only one causal basis per disposition” (ibid., p. 133). In addition: “Causal bases can be either dispositional or non-dispositional [properties]” (ibid., p. 132). For instance, it may be oftentimes said that the fragility of this glass has as causal basis some molecular structure (which is non-dispositional) of the glass [3][5]. On her view, this fragility has also the *fragility itself* as causal basis because it is causally relevant to its realization, namely the breaking of the glass.

As for the distinctness thesis, its denial by McKitrick is motivated to accommodate what she calls “ungrounded dispositions”: (epistemically and metaphysically possible) dispositions of fundamental physical entities (possibly subatomic particles) that have no non-dispositional causal basis. Because even ungrounded dispositions have some causal basis (i.e., themselves), the causal thesis would indicate that all dispositions in dispositional pluralism have in common their specific causal import under the name of their causal bases.

McKitrick’s view of causal bases of dispositions is coherent and it would be philosophically tenable, but there may be some concerns as to its direct import to our project to give a unifying perspective on realizable entities in formal ontology. First, her usage of the term “causal basis” could confuse normal ontology developers because a disposition can be its own causal basis. Surely, she says that this claim: “is not to say that a disposition causally explains itself, but only that it causally explains its manifestation [realization]” ([14], p. 133). However, we would have good reason to preserve the term “basis” in its intuitive and narrow sense that excludes something being a basis (whether causal or not) of itself, as is exemplified by the material basis (“disorder”) of a disease in the BFO-compliant OGMS dispositional account of diseases [21].

Second, the possibility of ungrounded dispositions would clearly deny the distinctness thesis, but this consequence will be of little usefulness in formal ontology because this field mostly deals with ordinary dispositions (e.g., fragility) that have non-dispositional causal bases. At the same time, the idea of ungrounded dispositions may be of potential practical value. Williams [27] propounds a domain-specific conception of ungrounded dispositions (to wit, of “powers” in his terms): a given (scientific) domain ascribes powers to entities that can be thought to be fundamental to the domain. He is skeptical of the philosophical plausibility of this view, but it can be deployed so that each domain ontology can be equipped with an associated ontology of what we may call “domain-relative ungrounded dispositions”: very roughly, dispositions in a given domain whose non-dispositional causal bases would belong to other domains. In this respect, McKitrick’s original notion of ungrounded disposition will serve as a starting point for developing its potentially useful domain-relative versions.

All these considerations can lead to the following formal specification of the causal import of realizable entities. This stipulates the generalization of the notion of a *categorical basis* [3][5] of a disposition (not necessarily in McKitrick’s pluralist sense of the term), through our reinterpretation, to a categorical basis of a realizable entity: a quality or a sum of qualities of the bearer of a realizable entity such that the quality (sum) makes the realizable entity causally relevant to its realization. Following the Williams[27]-inspired argument that McKitrick’s ungrounded dispositions are intrinsic because their causal import must stem only from their bearers, we can define an ungrounded disposition as an intrinsic disposition that has no categorical basis (d1):

d1 $\mathrm{UND}(x) =_{\mathrm{def.}} \mathrm{IND}(x) \wedge \neg\exists y\, \mathrm{HCB}(x,y)$

This will allow us to propose our (albeit weakened) “realizable counterpart” of the aforementioned causal thesis. That is to say, all other realizable entities than ungrounded dispositions have a categorical basis (a1):

a1 $\mathrm{REA}(x) \wedge \neg\mathrm{UND}(x) \rightarrow \exists y\, \mathrm{HCB}(x,y)$

Note that, as with causal bases of dispositions in McKitrick’s theory, one realizable entity may have multiple categorical bases.

3.3. Extrinsic Dispositions in Dispositional Pluralism

We are exploring realizable entities with recourse to a dispositional lens that is theoretically underpinned by dispositional pluralism. Here we will focus on one non-standard kind of dispositions only, namely on extrinsic dispositions. An extrinsic disposition is a disposition that exists (at least partially) in virtue of the way the world that is external to the bearer is. McKitrick’s examples of extrinsic dispositions include vulnerability (the disposition to be harmed if attacked), visibility (the disposition to be seen when someone looks towards it), weight (the disposition to depress a properly constructed scale relative to a local gravitational field, following Yablo [28]), and mass (the disposition to produce a gravitational force which is generated by its immersion in the Higgs field, following Bauer [29]).[3] As we said in Section 3.1, extrinsic dispositions would be externally grounded realizable entities in BFO. At the same time, we leave open whether the BFO internally/externally distinction in groundedness is to be (re)interpreted in terms of the intrinsic/extrinsic property distinction. (UFO [9][10] characterizes dispositions as a subtype of “intrinsic moments”, where moments correspond approximately to specifically dependent continuants in BFO.)

We deploy the following exemplar of extrinsic dispositions which can be attributed to Shoemaker’s [30] key/door example. Imagine this key (say $\mathbf{key_1}$) and this lock (say $\mathbf{lock_2}$) such that $\mathbf{key_1}$ opens $\mathbf{lock_2}$. Consider the realizable entity (say $\mathbf{re_1}$) of $\mathbf{key_1}$ to be realized in a process of the type *lock*$_2$*-opening-by-key*$_1$ and the realizable entity (say $\mathbf{re_2}$) of $\mathbf{lock_2}$ to be realized in a process of the same type. From a pluralist point of view, $\mathbf{re_1}$ and $\mathbf{re_2}$ are extrinsic dispositions because they are borne in virtue of the existence of $\mathbf{lock_2}$ and $\mathbf{key_1}$, respectively, as is indicated by Shoemaker’s [30] discussion of the key-door example in terms of Geach’s [31] notion of “mere Cambridge change” (which is, roughly, a change that does not involve any intrinsic change).

Extrinsic dispositions constitute a crucial group of externally grounded realizable entities because they are of great value for ontological modeling, above all of entities with environmental and social dimensions. (Note that, relatedly, intrinsic dispositions are equally important, as they are such paradigmatic dispositions as fragility and

---

[3] BFO takes mass to be an exemplar of its category of *quality* (see Section 2). This means that dispositional pluralism might possibly cover some kinds of qualities in BFO. To address this issue will require careful consideration of the BFO distinction between qualities and realizable entities, or more generally the general ontological distinction [13] between categorical and dispositional properties.

solubility.) For that matter, some dispositions in preceding formal-ontological work would prove to be extrinsic dispositions (rather than dispositions in BFO or UFO): for instance, Barton et al.'s [32] idea of a disposition with an "existential condition", namely a disposition which depends existentially on something that is external to the bearer.

For a concrete example, McKitrick [23] characterizes a gender as a cluster of behavioral and extrinsic dispositions (see Toyoshima et al.'s [24] work on this line of ontological representation of gender). To take another one, Turvey's [33] dispositional account of Gibson's [34] notion of affordance (roughly, what the environment "offers" agents) entails an ontological commitment to extrinsic dispositions, as is explained by Vetter [35]. To illustrate this, an affordance of this staircase is its disposition to support John as he moves upward (or downward) using the staircase and this affordance is an extrinsic disposition because it exists in virtue of John, who is not part of the bearer of this affordance (namely the staircase). Moreover, Turvey's affordances within the environment are always coupled with associated dispositions (which he calls "effectivities") of agents. Affordances and effectivities are both extrinsic dispositions and they can be indeed formalized by analogy with $\mathbf{re_1}$ and $\mathbf{re_2}$ [36].

We will consider two formal ways of explicating extrinsic dispositions. Firstly, they are in nature, in some sense, "derivative" of some intrinsic dispositions. To illustrate this, $\mathbf{re_1}$ "derives from" the intrinsic disposition (say $\mathbf{Re_1}$) of $\mathbf{key_1}$ to open any instance (e.g., $\mathbf{lock_2}$) of the type $Lock_2$, and $\mathbf{re_2}$ "derives from" the intrinsic disposition (say $\mathbf{Re_2}$) of $\mathbf{lock_2}$ to open any instance (e.g., $\mathbf{key_1}$) of the type Key_1 [36]. Williams [27] contends that an extrinsic disposition *depends* (existentially) on some intrinsic disposition in the sense that without the latter, the former would not exist (although, to wit, he uses the term "power"). Because BFO already has several ontological dependence relations, we introduce the "existential reliance" relation (REON for "relies on") between an extrinsic disposition and an intrinsic disposition to forestall terminological confusion: REON($\mathbf{re_1}$, $\mathbf{Re_1}$) and REON($\mathbf{re_2}$, $\mathbf{Re_2}$) hold. If REON($\mathbf{d_1}$, $\mathbf{d_2}$), we call $\mathbf{d_2}$ an "intrinsic dependee" of $\mathbf{d_1}$. Williams's claim can be then formalized as follows (a2):

a2 $\mathrm{EXD}(x) \land \mathrm{INH}(x,y) \rightarrow \exists z\, (\mathrm{IND}(z) \land \mathrm{INH}(z,y) \land \mathrm{REON}(x,z))$

We do not assume that REON is functional: an extrinsic disposition can have several intrinsic dependees. Suppose for example that $\mathbf{key_1}$ can open instances of $Lock_2$, as well as instances of $Lock_2'$, where $Lock_2$ is different from $Lock_2'$ and $\mathbf{lock_2}$ is an instance of both $Lock_2$ and $Lock_2'$. Then, $\mathbf{re_1}$ relies not only on $\mathbf{Re_1}$ (the intrinsic disposition to open instances of $Lock_2$), but also on $\mathbf{Re_1}$' (the intrinsic disposition to open instances of $Lock_2'$).

The REON relation can be logically constrained by means of disposition-related relations [3][5]. First of all, every realization and trigger of an extrinsic disposition is a realization and trigger of any intrinsic dependee thereof, respectively (a3, a4):

a3 $\mathrm{REON}(x,y) \land \mathrm{REAL}(x,z) \rightarrow \mathrm{REAL}(y,z)$
a4 $\mathrm{REON}(x,y) \land \mathrm{HTR}(x,z) \rightarrow \mathrm{HTR}(y,z)$

Note that the reciprocal does not hold: some realizations and triggers of an intrinsic dependee of an extrinsic disposition are not realizations and triggers of this extrinsic disposition, respectively. For example, if $\mathbf{lock_2}$' is an instance of $Lock2$ but different from $\mathbf{lock_2}$, then $\mathbf{Re_1}$ – in contradistinction with $\mathbf{re_1}$ – can be triggered by $\mathbf{key_1}$ turning into $\mathbf{lock_2}$', and realized in $\mathbf{key_1}$ opening $\mathbf{lock_2}$'.

We can also think that every categorical basis of an intrinsic dependee of an extrinsic disposition is a categorical basis of the extrinsic disposition (a5):

a5 $\mathrm{REON}(x,y) \wedge \mathrm{HCB}(y,z) \rightarrow \mathrm{HCB}(x,z)$

Note that the reciprocal does not hold: a categorical basis of an extrinsic disposition is not a categorical basis of an intrinsic dependee of it. For example, $\mathbf{re}_1$ has as categorical basis some features of $\mathbf{lock}_2$, whereas $\mathbf{Re}_1$ does not (all its categorical bases are features of $\mathbf{key}_1$).

One may think that an extrinsic disposition has also as its categorical basis something that is external to the bearer, as is illustrated by Barton et al.'s [32] idea of an "external base" of extrinsic dispositions. For instance, $\mathbf{re}_1$ and $\mathbf{re}_2$ may seem to have as their categorical bases some geometric structure of $\mathbf{lock}_2$ and $\mathbf{key}_1$, respectively. According to Contessa [37], however, extrinsic dispositions are a counterexample to the Intrinsic Dispositions Thesis ("All dispositions are intrinsic"), but its falsity does not entail the Intrinsic Bases Thesis ("The causal (categorical) bases of all dispositions are intrinsic"). We leave this issue open for future investigations.

Secondly, extrinsic dispositions may be further elucidated by dint of what may be called their "systemic view". The pivotal idea is that an extrinsic disposition exists within a system that is composed of its bearer and other objects. This systemic account of extrinsic dispositions may not be espoused by McKitrick [14] herself, but it seems to be propounded by Turvey's [33] dispositional approach to affordances (see Toyoshima & Barton's [36] detailed analysis). Vetter [38] provides its explicit formulation in terms of the notion of *potentiality*. Although she distinguishes potentialities from dispositions, McKitrick points out that dispositional pluralism can understand potentialities as a subtype of dispositions. We will below present a dispositional reinterpretation of Vetter's systemic approach to extrinsic dispositions.

The thrust of Vetter's [38] argument about extrinsic dispositions (or "extrinsic potentialities", in her terms) is that the possession of an extrinsic disposition by an object is both necessary and sufficient for the possession of a *joint disposition* ("joint potentiality" in her terms) by a system composed of this object and others. (Note that a joint disposition would be an intrinsic disposition that is borne by multiple objects "together".) In more detail: whenever an object bears an extrinsic disposition, this disposition is "fully grounded" (in her terms) in a joint disposition which is borne by a system composed of this object and others; and whenever a system composed of a number of objects bears a joint disposition, each of the objects thereby bear an extrinsic disposition which is fully grounded in that joint disposition. For instance, $\mathbf{key}_1$ and $\mathbf{lock}_2$ have extrinsic dispositions $\mathbf{re}_1$ and $\mathbf{re}_2$, respectively, in virtue of the fact that the "$\mathbf{key}_1$ & $\mathbf{lock}_2$ system" bears some joint disposition (say $\mathbf{re}_3$) that fully grounds $\mathbf{re}_1$ and $\mathbf{re}_2$, and *vice versa* (see also Toyoshima & Barton's [36] similar discussion in examining Turvey's [33] dispositional account of affordances and effectivities). Like $\mathbf{re}_1$ and $\mathbf{re}_2$, $\mathbf{re}_3$ can be realized by a process of the kind *$lock_2$-opening-by-key_1*.

To formalize this Vetter-style systemic account of extrinsic dispositions will require specifying the relationship between an extrinsic disposition and a joint disposition. Taking a cue from Toyoshima & Barton [36], we will employ the "add-part_of relation" (ADP) [4]. This add-parthood relation represents the *add*itive character of dispositions: for example, the solubility of this whole tablet has two add-parts, namely the solubility of the left half of the tablet and the solubility of the right half. Given the simplifying assumption that a system is a mereological sum of objects (refer to Röhl [39] for more thoughts), an extrinsic disposition can be seen as an add-part of a *joint* disposition, as $\mathbf{re}_1$ and $\mathbf{re}_3$ (or $\mathbf{re}_2$ and $\mathbf{re}_3$) satisfy the three axioms [4] characterizing add-parthood:

- The bearers of $\mathbf{re}_1$ and $\mathbf{re}_2$ (i.e., $\mathbf{key}_1$ and $\mathbf{lock}_2$) are (proper) parts of the bearer of $\mathbf{re}_3$ (i.e., the sum of $\mathbf{key}_1$ and $\mathbf{lock}_2$).
- If $\mathbf{re}_3$ is realized in a process of $\mathbf{key}_1$ opening $\mathbf{lock}_2$, then both $\mathbf{re}_1$ and $\mathbf{re}_2$ are realized in a part of this process (i.e., this very process).
- If $\mathbf{re}_3$ is triggered by a process of $\mathbf{key}_1$ pivoting in $\mathbf{lock}_2$, then both $\mathbf{re}_1$ and $\mathbf{re}_2$ are triggered by a part of this process (i.e., this very process).

A central tenet of a systemic theory of extrinsic dispositions is that for every extrinsic disposition x that is borne by an object y, there exist a joint disposition z that is borne by a system composed of x and other objects ($w_1, \dots, w_n$) that each bear an extrinsic disposition ($v_1, \dots, v_n$), such that x, $v_1, \dots, v_n$ are all add-parts of z (z is a joint disposition for x, for v_1, …, for v_n). Let SUM be a (n+2)-ary relation such that SUM(x, $y_1, \dots, y_{n+1}$) means (where n stands for a natural number that is at least 1): “x is a mereological sum of $y_1, \dots, y_{n+1}$”. This claim can be formalized as follows (a6), although it may be undefinable in first-order logic owing to the arbitrary length of sequences:

$$\textbf{a6}\quad \mathrm{EXD}(x) \wedge \mathrm{INH}(x,y) \rightarrow \exists z,u\ \mathrm{IND}(z) \wedge \mathrm{ADP}(x,z) \wedge \mathrm{INH}(z,u) \wedge \\ \exists v_1,\dots,v_n, w_1,\dots,w_n\ \ \mathrm{SUM}(u, y, w_1, \dots, w_n) \wedge \\ \bigwedge_{1 \leqq i \leqq n} (\mathrm{EXD}(v_i) \wedge \mathrm{INH}(v_i, w_i) \wedge \mathrm{ADP}(v_i, z))$$

We can illustrate (a6) with a puzzle made of three pieces (say $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$). $\mathbf{p}_1$ has the extrinsic disposition $\mathbf{re}_{\mathbf{p1}}$ to be joined with $\mathbf{p}_2$ and $\mathbf{p}_3$. By (a6), there is a joint disposition $\mathbf{d}_{\mathbf{p1\text{-}3}}$ that has $\mathbf{re}_{\mathbf{p1}}$ as add-part and whose bearer is the sum of $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$. Then, $\mathbf{p}_2$ (or $\mathbf{p}_3$, respectively) has an extrinsic disposition $\mathbf{re}_{\mathbf{p2}}$ (or $\mathbf{re}_{\mathbf{p3}}$, respectively) to be joined with $\mathbf{p}_1$ and $\mathbf{p}_3$ (or $\mathbf{p}_1$ and $\mathbf{p}_2$, respectively) such that $\mathbf{re}_{\mathbf{p2}}$ and $\mathbf{re}_{\mathbf{p3}}$ are also add-parts of $\mathbf{d}_{\mathbf{p1\text{-}3}}$.

To recapitulate briefly, an extrinsic disposition (borne by, say, **b**) has one or more intrinsic dependees (which are intrinsic dispositions of **b**), and is an add-part of a joint disposition, which is intrinsic and inheres in a system that has **b** as part.

4. Discussion: Applying our Dispositional Lens for Realizable Entities

4.1. Functions in BFO

At present, BFO characterizes functions as dispositions of bearers with a specific kind of historical development [40], although controversy exists as to the validity of the dispositional identification of functions [12][41]. In more details, a function is a disposition that its bearer possesses in virtue of its having a certain physical makeup because of how it came into being, either through evolution (when the bearer is a natural biological entity) or intentional design (when the bearer is an artifact).

Our dispositional framework for realizable entities can help to discern two kinds of dispositions that can be intuitively understood as functions, such as:

- the function (disposition) $\mathbf{f}_1$ of this heart to provide blood for human bodies in general, to wit, for any instance of the type *Human body*;
- the function (disposition) $\mathbf{f}_2$ of this heart to provide blood for this particular human body (say, Nancy’s body).

Note that $\mathbf{f_1}$ is intrinsic but $\mathbf{f_2}$ is extrinsic because it exists in virtue of Nancy's body; in other words, $\mathbf{f_2}$ exists only with respect to her bodily system. Given our hypothesis that extrinsic dispositions would be outside the BFO category of disposition, the BFO theory of functions can account for $\mathbf{f_1}$, but not for $\mathbf{f_2}$. One possible interpretation is that, because $\mathbf{f_2}$ relies on $\mathbf{f_1}$, functions in BFO can be elucidated as functions that are intrinsic dependees (e.g., $\mathbf{f_1}$) of the kind of realizable entities (e.g., $\mathbf{f_2}$) that might also count informally as functions but that would not be classified as functions by the current version of BFO.

In this way, extrinsic dispositions can be expected to shed light on a general ontology of functions. For example, it will be interesting to use extrinsic dispositions (as they consist in being within some system having their bearers as component) to analyze so-called "causally contribution theories" of functions [42]: roughly, a function is the associated causal role within a system that has the function bearer as component.

4.2. Roles in BFO

We will finally consider roles in BFO, partly because they remain currently largely unexplored, partly because their study will make a practical contribution e.g., to an enhanced representation of social roles in the BFO-compliant Ontology of Medically Related Social Entities (OMRSE) [43]. To expand our dispositional approach to roles, it will be necessary to show how role terms can be well specified in terms of dispositions. Examples of canonical role terms include "student", "president", and "money". Among other things, students are frequently discussed in prior work, as Boella et al. [44] say that they are a "rather simple" example of roles. On closer examination, however, students turn out to be ontologically multifaceted [20]. For one thing, a student is a paradigmatic example of "social roles" [45] and, as Loebe [46] says, their full-fledged analysis will demand a solid theory of social ontology [47], which has been actively researched in formal ontology [48] and lies outside the scope of this paper (see Toyoshima's [20] discussion on deontic and normative aspects of social roles).

Accordingly, we will begin with some non-social role. As a matter of fact, BFO recognizes non-social examples of roles such as "the role of a stone in marking a boundary" ([7], p. 100). In particular, we will focus on the role term "catalyst" in the sense of being a substance that makes a chemical reaction happen faster without being changed itself. One might wonder whether a catalyst should be analyzed as a role in BFO, but it will be illuminating to consider from our dispositional viewpoint why catalysts are a somewhat controversial example of roles.[4] Suppose that this amount of manganese (say $\mathbf{m_1}$) significantly speeds up the process of this amount of hydrogen peroxide (H_2O_2), say $\mathbf{hp_2}$, turning into water (H_2O) and oxygen (O_2). We can also say, based on BFO, that $\mathbf{m_1}$ has a role (say $\mathbf{role_{m1}}$) of being a catalyst to be realized in the decomposition of $\mathbf{hp_2}$.

Let us consider $\mathbf{role_{m1}}$ from our dispositional point of view. First of all, it will be a natural starting point to ask whether at least some (if not) roles in BFO can be seen as extrinsic dispositions, since roles and extrinsic dispositions are both a subtype of externally grounded realizable entities. An affirmative answer to this question may be supported by preceding work on roles. As Boella et al. [44] say, for instance, Baldoni et al. [49] espouse the view that a role (in its general sense) can be understood in terms of

---

[4] One consideration in favor of a role view of catalysts could be provided by Chemical Entities of Biological Interest (ChEBI), a database and ontology of molecular entities focusing on "small" chemical compounds. ChEBI says that a catalyst is a "chemical role": "A role played by the molecular entity or part thereof within a chemical context" (https://www.ebi.ac.uk/chebi/chebiOntology.do?chebiId=CHEBI:35223; last accessed on July 19, 2021).

an aggregate of affordances. To borrow their example, Jack's role of being a user of a printer can be represented with what this printer affords Jack, such as printed text on paper. When combined with Turvey's [33] dispositional account of affordances (see Section 3.3), this affordance-based theory of roles would favor the conception of roles as a subtype of extrinsic dispositions [50]: Jack's user role under discussion would be his effectivity of printer-user that is inextricably linked with the affordance of the printer.

In this direction, we can think of **role**$_{\mathbf{m1}}$ as an extrinsic disposition to accelerate the decomposition of **hp**$_{\mathbf{2}}$ when **m**$_{\mathbf{1}}$ meets **hp**$_{\mathbf{2}}$. For that matter, Vetter [38] cites this catalyst example in discussing joint dispositions ("joint potentialities" in her terms). Since an extrinsic disposition has some intrinsic dependee (see Section 3.3), it is reasonable to consider the intrinsic disposition (say **d**$_{\mathbf{m1}}$) of **m**$_{\mathbf{1}}$ to catalyze hydrogen peroxide in general, to wit, any instance of the type *Hydrogen Peroxide*. This analysis of **role**$_{\mathbf{m1}}$ will highlight the importance of the disambiguation of the term "catalyst", for **d**$_{\mathbf{m1}}$ may well be described as **m**$_{\mathbf{1}}$'s realizable entity of being a catalyst for the decomposition of hydrogen peroxide, just as **role**$_{\mathbf{m1}}$ may be described as **m**$_{\mathbf{1}}$'s realizable entity of being a catalyst for **hp**$_{\mathbf{2}}$. To put it precisely, our claim is that a catalyst role would be an extrinsic disposition.

We contend that this dispositional understanding of non-social roles (including catalysts) could be generalized to social roles, together with some auxiliary theories (e.g., social ontology). To provide a pointer to future inquiry, consider Alice's social role (say **doctor**$_{\mathbf{A}}$) of being a doctor. First of all, assuming that being able to treat a person is a minimal element of being a doctor, Alice would cease to be a doctor when she is entirely incapable of treating a person. This claim can be captured when **doctor**$_{\mathbf{A}}$ is analyzed in terms of Alice's intrinsic disposition to treat a person.

Moreover, the current dispositional view of social roles may have the potential to tackle the issue of contexts for social roles. The notion of context is generally reckoned to be germane to roles in the sense that roles would cease to exist when their contexts do [19][20][45]. In particular, as Loebe [46] says, the challenge of considering social roles is partly due to the intricacy of their contexts (e.g., schools for student roles). One possible hypothesis is that, provided that social roles can be dispositionally approached, their contexts would be systems with respect to which associated extrinsic dispositions exist. To illustrate this, consider Alice's role (say **treater**$_{\mathbf{A}}^{\mathbf{B}}$) of treating Bob, as it is closely linked with **doctor**$_{\mathbf{A}}$. A context for **treater**$_{\mathbf{A}}^{\mathbf{B}}$ would be the "Alice & Bob system" which is composed of Alice, Bob, a joint disposition which fully grounds **treater**$_{\mathbf{A}}^{\mathbf{B}}$ and also Bob's role (say **treatee**$_{\mathbf{B}}^{\mathbf{A}}$) of being a person of Alice's treatment (note the complex ontological nature of a system [39], which is composed, in a sense that we do not analyze here, of independent continuants and realizable entities).

To be sure, there is a non-trivial difference between the Alice & Bob system (which is a context for **treater**$_{\mathbf{A}}^{\mathbf{B}}$) and a context for **doctor**$_{\mathbf{A}}$, possibly the hospital to which Alice belongs. To fill this gap will require scrutiny of many social roles that are intimately connected with **doctor**$_{\mathbf{A}}$: e.g., others doctors, patients, and nurses in Alice's hospitals. It will be also necessary to take into account extrinsic dispositions that shape the social import of **doctor**$_{\mathbf{A}}$, such as the disposition (*à la* Donohue [51]) of the hospital committee to sanction Alice when she fails to follow a designated guideline for treatment. In this way, our dispositional framework for realizable entities would form the basis for a full ontological analysis of social roles and their contexts.

Finally, we emphasize the importance of the disambiguation of the term "role" and social role terms (e.g., "student") because, when they are rather difficult to analyze in our dispositional approach to roles in BFO, they may be better interpreted in terms of other BFO categories than realizable entities. In effect, Arp et al. [7] state: “The term ‘role’ can (…) be used in a different sense in contexts such as Jane’s being the seventh person to fill the role of director of this institute (…). ‘Role’ in this sense is being used to designate what BFO calls a *generically dependent continuant*” (ibid., pp. 100-101). It is worth registering a possible linkage between roles that are a subtype of generically dependent continuants and Brochhausen et al.’s [52] idea of “socio-legal, generically dependent continuants” which come into existence through declarations and which are concretized in roles in the BFO sense of the term.

5. Conclusion

The principal goal of this paper was to launch a systematic investigation into realizable entities, as they figure in a large variety of domains. To achieve it, we adopted a disposition-centered methodology for considering realizable entities in the BFO upper ontology that is theoretically underwritten by McKitrick’s [14] dispositional pluralism. In particular, we examined extrinsic dispositions because they may encompass a wide range of entities, including gender [23][24]. We also discussed functions and roles in BFO through our “dispositional lens” for realizable entities. Those first important steps towards a systemization of realizable entities will need to be completed by future works. For example, further investigation is warranted into our systemic account of extrinsic dispositions, especially into Vetter’s [38] “full grounding” relation between dispositions.

References

[1] Guarino N, Guizzardi G. Relationships and events: towards a general theory of reification and truthmaking. In: Proceedings of AI*IA 2016. LNCS, vol. 9097. Springer; 2016. p. 237-49.
[2] Guarino N. BFO and DOLCE: so far, so close…. Cosmos + Taxis. 2016;4(4):10-8.
[3] Röhl J, Jansen L. Representing dispositions. J Biomed Semant. 2011 Aug;2(Suppl 4):S4.
[4] Barton A, Jansen L, Ethier JF. A taxonomy of disposition-parthood. In: Proceedings of JOWO2017. CEUR Workshop Proceedings, vol. 2050; 2017. p. 1-10.
[5] Barton A, Grenier O, Jansen L, Ethier JF. The identity of dispositions. In: Proceedings of FOIS2018. Amsterdam: IOS Press; 2018. p. 113-25.
[6] Toyoshima F. Natural necessity: an introductory guide for ontologists. Appl Ontol. 2020 Feb;15(1):61-89.
[7] Arp R, Smith B, Spear AD. Building ontologies with Basic Formal Ontology. MIT Press; 2015. 248 p.
[8] ISO/IEC PRF 21838-2.2 [internet]. Information technology - Top-level ontologies (TLO) - Part 2: Basic Formal Ontology (BFO); 2020 Mar. Available from: https://www.iso.org/standard/74572.html
[9] Guizzardi G. Ontological foundations for structural conceptual models. [dissertation]. Centre for Telematics and Information Technology: University of Twente; 2005. 416 p.
[10] Guizzardi G, Wagner G, Almeida JAP, Guizzardi RSS. Towards ontological foundations for conceptual modeling: the unified foundational ontology (UFO) story. Appl Ontol. 2015;10(3-4):259-71.
[11] Jansen L. Tendencies and other realizables in medical information sciences. The Monist. 2007 Oct;90(4): 534-54.
[12] Röhl J, Jansen L. Why functions are not special dispositions: an improved classification of realizables for top-level ontologies. J Biomed Semantics. 2014 Jun 2;5:27.
[13] Choi S, Fara M. Dispositions. In: Zalta EN, editor. The Stanford encyclopedia of philosophy. Spring 2021 ed. Available from: https://plato.stanford.edu/archives/spr2021/entries/dispositions/
[14] McKitrick J. Dispositional pluralism. Oxford University Press; 2018. 261 p.
[15] Smith B, Ceusters W. Ontological realism: a methodology for coordinated evolution of scientific ontologies. Appl Ontol. 2010 Nov;5(3-4):139-88.
[16] Wachter, D von. Dinge und Eigenschaften: Versuch zur Ontologie. Verlag J.H. Röll; 2000. 252 p.
[17] Guarino N, Welty C. An overview on OntoClean. In: Staab S, Studer R, editors. Handbook on ontologies.

2nd edition. Springer; 2009 p. 201-20.

[18] Seyed AP, Shapiro SC. A method for evaluating ontologies - introducing the BFO-rigidity decision tree wizard. In: Proceedings of FOIS2012. Amsterdam: IOS Press; 2012. p. 191-204.

[19] Mizoguchi R, Galton A, Kitamura Y, Kozaki K. Families of roles: a new theory of occurrent-dependent roles. Appel Ontol. 2015;10(3-4):367-99.

[20] Toyoshima F. Roles and their three facets: a foundational perspective. Appl Ontol. 2021 Apr;16(2):161-92.

[21] Scheuermann RH, Ceusters W, Smith B. Toward an ontological treatment of disease and diagnosis. In: Proceedings of the 2009 AMIA Summit on translational bioinformatics. 2009. p. 116-20.

[22] Toyoshima F. Barton A, Grenier O. Foundations for an ontology of belief, desire and intention. In: Proceedings of FOIS2020. Amsterdam: IOS Press. p. 140-54.

[23] McKitrick JA. A dispositional account of gender. Philos Stud. 2015;172:2575-89.

[24] Toyoshima F, Barton A, Fabry P, Ethier JF. Representing gender in ontologies: a dispositional perspective. In: Proceedings of ICBO2021. CEUR Workshop Proceedings; 2021. Forthcoming.

[25] Marshall D, Weatherson B. Intrinsic vs. extrinsic properties. In: Zalta EN, editor. The Stanford encyclopedia of philosophy. Spring 2018 ed. Available from: https://plato.stanford.edu/archives/spr2018/entries/intrinsic-extrinsic/

[26] Prior E, Pargetter R, Jackson F. Three theses about dispositions. Am Philos Q. 1982 Jul;19(3):251-57.

[27] Williams NE. The powers metaphysic. Oxford University Press; 2019. 288 p.

[28] Yablo S. Intrinsicness. Philos Topics. 1999 Jan;26(1):479-505.

[29] Bauer WA. An argument for the extrinsic grounding of mass. Erkenn. 2011 Jan;74:81-99.

[30] Shoemaker S. Causality and properties. In: van Inwagen P, editor. Time and cause. Springer; 1980. p. 109-35.

[31] Geach PT. God and the Soul. St Augustine's Press; 1969. 138 p.

[32] Barton A, Rovetto R, Mizoguchi R. Newtonian forces and causation: a dispositional account. In: Proceedings of FOIS2014. Amsterdam: IOS Press; 2014. p. 157-70.

[33] Turvey MT. Affordances and prospective control: an outline of the ontology. Ecol Psychol. 1992;4:173-87.

[34] Gibson JJ. The ecological approach to visual perception: classic edition. Psychology Press; 2014. 316 p.

[35] Vetter, B. Perceiving potentiality: a metaphysics for affordances. Topoi. 2020:39:1177-91.

[36] Toyoshima F, Barton A. A formal representation of affordances as reciprocal dispositions. In: Proceedings of TriCoLore2018. CEUR Workshop Proceedings, vol. 2347; 2019. p. 1-9.

[37] Contessa G. Do extrinsic dispositions need extrinsic causal bases? Philos Phenomenol Res. 2012 May;84(3):622-38.

[38] Vetter B. Potentiality: from dispositions to modality. Oxford: Oxford University Press; 2015. 335 p.

[39] Röhl J. Mechanisms in biomedical ontology. J Biomed Semantics. 2012;3(Suppl 2):S9.

[40] Spear AD, Ceusters W, Smith B. Functions in Basic Formal Ontology. Appl Ontol. 2016 Jun;11(2):103-28.

[41] Jansen L. Functions, malfunctioning, and negative causation. In: Christian A, Hommen D, Retzlaff N, Schurz G, editors. Philosophy of science. Springer; 2018. p. 117-35.

[42] Cummins R. Functional analysis. J Philos. 1975 Nov;72(20):741-65.

[43] Hicks A, Hanna J, Welch D, Brochhausen M, Hogan WR. The ontology of medically related social entities: recent developments. J Biomed Semantics. 2016 Jul;7:47.

[44] Boella G, van der Torre L, Verhagen H. Roles, an interdisciplinary perspective. Appl Ontol. 2007;2:81-8.

[45] Masolo C, Vieu L, Bottazzi E, Catenacci C, Ferrario R, Gangemi A, Guarino N. Social roles and their descriptions. In: Proceedings of KR2004. 2004. p. 267-77.

[46] Loebe F. Abstract vs. social roles - towards a general theoretical account of roles. Appl Ontol. 2007;2:127-58.

[47] Searle JR. The construction of social reality. Simon and Schuster; 1995. 256 p.

[48] Jansen L. Towards an upper-level ontology for the social domain. In: Proceedings of JOWO2019. CEUR Workshop Proceedings, vol. 2518; 2019. p. 1-11.

[49] Baldoni M, Boella G, Van Der Torre L. Modelling the interaction between objects: roles as affordances. In: Proceedings of KSEM2006. LNCS, vol. 4092. Springer; 2006. p. 42-54.

[50] Barton A, Toyoshima F, Ethier JF. Approaching roles with affordances. In: Proceedings of JOWO2020. CEUR Workshop Proceedings, vol. 2708; 2020. p. 1-14.

[51] Donohue B. Social ontology and social normativity. [dissertation]. University at Buffalo;2020. 170 p. Available from: https://philpapers.org/rec/DONSOA-3

[52] Brochhausen M, Almeida MB, Slaughter L. Towards a formal representation of document acts and the resulting legal entities. In: Svennerlind S, Almäng J, Ingthorsson R, editors. Johanssonian investigations: essays in honour of Ingvar Johansson on his seventieth birthday. Walter de Gruyter; 2013. p. 120-139.

II. Applications and Methods

This page intentionally left blank

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210373

Asymmetric Hybrids: Dialogues for Computational Concept Combination

Guendalina RIGHETTI [a], Daniele PORELLO [b], Nicolas TROQUARD [a], Oliver KUTZ [a], Maria M. HEDBLOM [c], Pietro GALLIANI [a]
[a] *Free University of Bozen-Bolzano, Italy*
[b] *University of Genova, Italy*
[c] *University of Bremen, Germany*

Abstract. When people combine concepts these are often characterised as “hybrid”, “impossible”, or “humorous”. However, when simply considering them in terms of extensional logic, the novel concepts understood as a conjunctive concept will often lack meaning having an empty extension (consider “a tooth that is a chair”, “a pet flower”, etc.). Still, people use different strategies to produce new non-empty concepts: additive or integrative combination of features, alignment of features, instantiation, etc. All these strategies involve the ability to deal with conflicting attributes and the creation of new (combinations of) properties. We here consider in particular the case where a Head concept has superior ‘asymmetric’ control over steering the resulting concept combination (or hybridisation) with a Modifier concept. Specifically, we propose a dialogical approach to concept combination and discuss an implementation based on axiom weakening, which models the cognitive and logical mechanics of this asymmetric form of hybridisation.

Keywords. Concept Combination, Hybridisation, Axiom Weakening, Dialogues, Compositionality

1. Introduction

Meredith: This is like a haunted coffeehouse thing?
Michael: No. Dwight is confusing you. That - it’s, it’s more of a disco.
Andy: It’s like a haunted disco.
Michael: ... with coffee but without the haunted.
Phyllis: It’s a combo dance house coffee bar.
Michael: It’s a daytime disco on the ground floor of an industrial office building.
Erin: It’s a cafe disco.
Michael: Exactly.
Kevin: So, like, a disco cafe?
Michael: Wha - No. No. Not even close.

The Office, Season 5, “Cafe Disco” [1]

The scene above demonstrates an interesting phenomenon. Namely, that concepts can be interpreted in different ways based on the weights of their attributes. The differentiation between a ‘disco cafe’ and a ‘cafe disco’ is determined by which of the two concepts has the more prominent role in the compound. While this difference might be quite intuitive for a native English speaker, it is a non-trivial problem to construct and explain in a formal setting. There exist different views of what concepts are and how they should be represented. The logic-based view aims to represent concepts in term of definitions or, more precisely, as sets of individually necessary and jointly sufficient conditions [2]. In this setting, the combination of two or several concepts is commonly understood in terms of set theoretic operations. This view presents advantages for classic knowledge representation, mostly because it offers a compositional and well-understood semantics that is in line with mainstream reasoning systems. Unfortunately, empirical evidence in psychology and cognitive science has shown that many concepts lack precise definitions, being subject to various degrees of indeterminacy as well as to typicality effects [3]. Moreover, as we will further discuss, a number of cognitive phenomena linked to concept combination are difficult to reconcile with a straightforward modelling of concepts using Boolean extensional logic [4].

This paper will analyse the case of "incompatible" combinations, based on the empirical research on *impossible combinations* [5, 6, 7] and hybridisation [8], focusing on *asymmetric combinations*. Impossible conceptual combination (e.g. the combination of Fish and Vehicle, of Furniture and Fruit,..) has been studied in the context of experimental psychology to investigate the flexibility and adaptability of concept meaning. If we look at concepts simply from an extensional point of view, when combining concepts without obvious similarities or shared features, the intersection will often be empty. Still, people use different strategies to produce creative non-empty concepts: alignment of features, instantiation, features emergence etc. These strategies involve the ability to deal with conflicting attributes and the creation of new properties: simply put, a sort of game of meaning negotiation.

In order to elucidate and model the cognitive and logical mechanics in this kind of asymmetric concept combination, we here propose a computational framework based on three essential ingredients:

(1) a computational model of concept combination taking into account cognitive aspects [9];
(2) formal approaches based on axiom weakening [10];
(3) an agent-based dialogical implementation[1] combining (1) and (2) to simulate meaning negotiation and construction in the asymmetric combination as it is found in the literature on hybrid concepts [7, 8].

Our approach to hybrid concept combination is related to conceptual blending (e.g. [11]) and, in particular, the distinction between Head and Modifier is reminiscent of asymmetric amalgams [12]. Please note, however, that our approach does not rely on the identification of a shared structure between the two input spaces via anti-unification — i.e. it does not require the identification of a *generic space* to steer the combination process. In contrast, our focus is on the asymmetric roles of Head and Modifier in the combination process [6], the integration process, and the resulting hybrid aspects of the combined concept [7, 8].

A similar distinction between Head and Modifier concepts is used in [13], where the authors propose an algorithm for concept combination based on a

---

[1] The notion of dialogue here is quite abstract and the choice of the term "dialogue" is justified by the intention to point to the literature in multiagent systems and dialogical logics.

non-monotonic description logic of typicality. In contrast to our approach, which relies on axiom weakening, the proposed algorithm discards all the axioms of the Modifier which are inconsistent with the Head concept. The only constraint is that the combined concept should not be trivial, i.e., should not include *all* the features associated with the Head concept. It is not obvious how such a procedure could account for the kind of impossible combination proposed here.

The agent-based dialogical concept combination proposed in this paper is taking concept combination approaches further particularly regarding the so far rather monolithic debugging techniques for inconsistent blends, see [14].

More in line with our work is [15], which employs epistemic logic to negotiate the debugging of aligned ontologies. In terms of cognitive heuristics, the work proposed here goes beyond the plain distinction between Head and Modifier, and instead presents a model of steering the dynamics of cognitive concept combination as suggested by the results of Hampton [6].

2. Forms of Concept Combination: Hybridity and Impossibility

Knowledge Representation (KR) systems are usually characterised by their compositional behaviour. Compositionality is the principle according to in which any complex concept or expression is understood as a function of the parts it is composed by, plus a set of syntactic operations to combine them. This perspective became a cornerstone of classical logic and moved from there to be also a paradigm in description logic. In this setting, where concepts are essentially considered in terms of sets, the combination of two (or several) concepts is mostly understood in terms of set theoretic operations. The compound concept "Tool Weapon" would be understood as the intersection of the set of tools and the set of weapons (the component concepts). Compositionality is sometimes used to explain, at least in part, the ease and prolific ability by which humans create and understand new and meaningful phrases, arguably, part of its theoretical strength. In KR, in particular, it offers the advantage of having a clear and well understood semantics. Related to compositionality, one beneficial feature of many KR systems is attribute inheritance. Namely, for each class A in an ontology, the instances of sub-classes $B \sqsubseteq A$ would inherit all the attributes from the superclass. For combined concepts this would mean that what lies in the intersection of two concepts would inherit all the features normally associated to any conjunct (see [16] for a recent simulation study on inheritance illustrating the complexity of the issues involved).

The process of concept combination has been extensively studied in the field of cognitive science and experimental psychology. This led to several distinct accounts of concept combination, diverging widely from what is expressible in terms of intersections of sets [4, 8, 17, 18].

Hybridity. For instance, it is possible to distinguish between different kinds of combinations depending on whether we consider adjective-noun combinations or noun-noun combinations. Although, in simpler logical modellings, they are often treated in the same way, it is at the same time normally assumed that noun-noun combinations involve much more semantic change in the compound concept [18]. Looking at noun-noun combinations in English, two parts can be distinguished, the Head and the Modifier, depending on the syntactic construction of the compound (this has been extensively studied in Linguistics, see e.g. [19]). Considering again "Tool Weapon", the noun "weapon" would play here the role of the Head, whereas "tool" would be the Modifier. As the names suggest, the Head provides the base category of the combined concept, whilst the Modifier alters the attributes of the Head. This means that humans interpret "Weapon Tool" (e.g. a certain repair tool for the Avtomat Kalashnikova) significantly different from a "Tool Weapon" (e.g. James Bond's typical screwdriver-shaped flame thrower). Clearly, any formal system employing compositional and commutative conjunction for such purposes would not be able to distinguish the two cases. Accounting for the difference in attribute inheritance is an important but logically challenging problem.

According to Wisniewski [8], there exist at least three ways to interpret noun-noun combinations: 1) The first is the *relation-linking* interpretation, where some kind of relation between the components is highlighted (using Wisniewski's example [8, p. 168] a *robin snake* is a snake that eats robin). 2) The second is the *property* interpretation, where one or more properties of the Modifier noun apply to the Head concept (a *robin snake* is a snake with a red under-belly [8, p. 169]). 3) The third is called *hybridisation*, where the result of the combination corresponds essentially to a 'mesh-up' or 'blend' of both components.

We focus here on the third kind of combination interpretation, *Hybridisation*, and give a formal definition and computational account of it. In [8], the author refers to this last kind as a "combination of the two constituents [...] or a conjunction of the constituents" (p. 169). Conceptually, this corresponds to the combinations analysed in [4]. In [4]'s experiments, people were asked to interpret noun-noun combinations expressed with a that-clause (e.g. a Tool Weapon is expressed as a "Weapon that is also a Tool"). This was done to encourage people to think of the combination in conjunctive terms [6]. Hampton's experiments [4] are of particular interest because he analysed the combination of ordinary concepts in terms of a logical interpretation. He found that, although it was possible to identify predictable patterns in the relation between compound and components, people are often not consistent with the rules of set theory.[2]

Impossibility. In a series of experiments [5, 6, 7], Hampton asked people to combine concepts that usually would not be combined, leading to impossible, or at least imaginary, objects. The aim of the investigation was to analyse the underlying principles for concept combination in a setting free from bias and prior knowledge of the concepts involved in order to study how adaptable and flexible concepts are. In [6], people were presented with a list of concept pairs (e.g. "Furniture" and "Fruit", "Vehicle" and "Fish", "Bird" and "Kitchen Utensil", etc.). Then, they were asked to imagine the objects resulting from the combination (e.g. "a Vehicle which is also a Fish"), and to describe, or draw, the attributes they would expect such an object to have. If analysed in terms of set-theoretic operations, the intersection of the concepts involved would be the empty set, and the set of axioms associated to both component concepts would likely

---

[2] For a formal analysis of these Hampton phenomena in weighted logics, we refer to [20, 21].

be inconsistent. Still, subjects showed a great variability of strategies to solve incompatible combinations. In this context, the process of properties *alignment* is particularly interesting. In order to select the 'right' properties for the compounds, people try to align properties and functions of the two component concepts. This was also noticed by [8], which, particularly in relation to hybridisation, proposed a comparison and alignment model where the Head and Modifier concepts are first aligned, so that the properties of one concept are put in correspondence with the properties of the other, and then are compared to find connections. This process can then go in quite different directions. First, subjects can find some *alignable differences* [17]: once the two representation are aligned, it is possible to find differences wrt. some of the aligned properties (or dimensions). In [6, table 9], for instance, when people analysed the concept "a Fruit which is a kind of Furniture", they tended to align the skin of the fruit with the fabric of a sofa. Or, when asked about "a Vehicle which is a kind of Fish", they reasoned that both could move, or made the analogical mapping between fuel and food. This alignment process corresponds to either finding commonalities in the differences that helps to understand which properties of the Modifier are integrated into the Head concept, or to find strong incompatibilities between the concepts. For instance, subjects could notice that, whereas a piece of furniture is made to last, fruit is perishable. Likewise, while a vehicle is normally controlled, a fish is likely to be 'self-motivated'. In these cases, people react to incompatibilities producing new, or *emergent* attributes [6]. Emergent attributes are defined as features that were not listed as true neither of the Head concept, nor of the Modifier, but that appear, or emerge, for the combined concept.

Another strategy observed by Hampton in his experiments is the process of *instantiation*: when asked to combine two super-ordinate categories (such as Vehicle and Fish or Fruit and Furniture), people would find it easier to come up with a solution "instantiating" one of them to a more basic and well-known category (combining e.g. Boat and Fish, or Banana and Furniture). Moreover, some of the categories were more likely to be instantiated than others (for instance, the category of Vehicle was almost always instantiated, whereas this very rarely happened to the one of Fish). The phenomenon of instantiation does not have an obvious explanation, since a more general concept would pose fewer constraints on the combination. Likely this is due to the fact that basic categories are easier to be imagined and more familiar to subjects, with more concrete properties to combine [7].

Aside from all these strategies, it is possible to observe some asymmetries between the Head and Modifier even in the case of impossible combinations. Hampton [7] shows that the solutions elaborated from the subjects usually bear more similarity to the Head noun than to the Modifier. Also in the case of impossible combinations, subjects keep the Head noun as a base to be modified by means of the Modifier.

3. Concept Refinements and Axiom Weakening

We here consider ontologies as sets of formulae in an appropriate logical language with the purpose of describing a particular domain of interest. The choice of a specific logic used is not crucial to our general approach, but we here employ the well-known description logic $\mathcal{ALC}$ both for illustrating examples and in our experimental prototype implementation. Syntactically, the $\mathcal{ALC}$ concept language is based on two disjoint sets N_C and N_R of *concept names* and *role names*, respectively. The set of $\mathcal{ALC}$ *concepts* is generated by the grammar $C ::= A \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall R.C \mid \exists R.C$, where $A \in N_C$ and $R \in N_R$. We denote by $\mathcal{L}(N_C, N_R)$ the set of $\mathcal{ALC}$ concepts built over N_C and N_R. Axioms are *general concept inclusions* (GCIs) or *individual assertions*. GCIs are of the form $C \sqsubseteq D$, where C and D are concepts. Finite sets of GCIs are called *TBoxes* which constitute the general, terminological knowledge of the ontology. Finally, individual assertions are of the form $C(a)$ where C is a concept and a an individual name; such statements constitute *ABoxes*. We assume standard DL syntax and semantics [22]. By $\mathsf{sub}(C)$ we denote the set of subconcepts of C, defined recursively as usual. The set of subconcepts in an axiom $C \sqsubseteq D$ is $\mathsf{sub}(C \sqsubseteq D) = \mathsf{sub}(C) \cup \mathsf{sub}(D)$; also, $\mathsf{sub}(C(a)) = \mathsf{sub}(C)$, and $\mathsf{sub}(O)$ denotes the set of all the subconcepts of all the axioms in O.

Refinement operators [10, 23] have been used for ontology aggregation in [24]. Here, they will be instrumental for concept combination. Given the quasi-ordered set $\langle \mathcal{L}(N_c, N_R), \sqsubseteq \rangle$, a generalisation refinement operator satisfies $\gamma_O(C) \subseteq \{C' \in \mathcal{L}(N_c, N_R) \mid C \sqsubseteq_O C'\}$. A specialisation refinement operator satisfies $\rho_O(C) \subseteq \{C' \in \mathcal{L}(N_c, N_R) \mid C' \sqsubseteq_O C\}$. Generalisation refinement operators take a concept C as input and return a set of descriptions that are more general than C by taking an ontology O into account. A specialisation operator, instead, returns a set of more specific descriptions. We can now define the notion of axiom weakening. The set of weakenings of an axiom wrt. a reference ontology O is defined as follows. Given an inclusion axiom $C \sqsubseteq D$ of O, the set of (least) *weakenings* of $C \sqsubseteq D$ wrt. O, denoted by $g_O(C \sqsubseteq D)$, is the set of all axioms $C' \sqsubseteq D'$ such that $C' \in \rho_O(C)$ and $D' = D$ or $C' = C$ and $D' \in \gamma_O(D)$. Given an individual assertion $C(a)$ of O, the set of (least) *weakenings* of $C(a)$, denoted $g_O(C(a))$, is the set of all axioms $C'(a)$ such that $C' \in \gamma_O(C)$.

The proposal laid out in this paper can make use of any refinement operator. When specific refinement operators are needed, as e.g. in examples and the implementation, we will be using the refinement operators from [10].

4. Dialogues for Concept Combination

We assume two agents, h and m, are interacting, trying to build a consistent compromise ontology R describing a concept. Each agent has an ontology associated, O_h and O_m, describing their initial version of R. Moreover, they each have a preference orderings $\prec_h$ and $\prec_m$ over the axioms of their respective ontology. The preferences represent the importance of the axiom for describing the concept.

In the dialogue, the agents are proposing in turn axioms coming from their ontology to be added to the ontology under construction R, weakening them when necessary. This is inspired by an approach from [24] for ontology aggregation. When the axioms proposed by the agents turn out to render the devised ontology inconsistent, the axiom weakening procedure is called to solve that inconsistency. A dialogue protocol is described informally in Algorithm 1.

Algorithm 1 Combination($O_{init}, O_h, \prec_h, O_m, \prec_m, prob_h$)

```
$R \leftarrow O_{init}$
TreatedAxioms $\leftarrow \emptyset$
FinishedAgents $\leftarrow \emptyset$
ag $\leftarrow$ RandomPickOneAgent($prob_h$, {h,m})
while FinishedAgents $\neq$ {h,m} do
    if $\forall$ axiom $\varphi \in O_{\text{ag}}$: $\varphi \in$ TreatedAxioms or $R$ entails $\varphi$ then
        FinishedAgents $\leftarrow$ FinishedAgents $\cup$ {ag}
    else
        $\varphi \leftarrow$ FavoriteNextAxiom($\prec_{\text{ag}}, O_{\text{ag}}$, TreatedAxioms, $R$)  ▷ Favorite axiom of agent ag, not in TreatedAxioms, and not entailed by $R$
        TreatedAxioms $\leftarrow$ TreatedAxioms $\cup\{\varphi\}$
        while $R \cup \{\varphi\}$ is inconsistent do
            $\varphi \leftarrow$ RandomPickOneWeakening($g_R(\varphi)$)
        $R \leftarrow R \cup \{\varphi\}$
        ag $\leftarrow$ RandomPickOneAgent($prob_h$, {h,m})
return $R$
```

The algorithm takes a few parameters: an initial ontology O_{init}, an ontology O_i for each agent $i \in \{h, m\}$, a (strict) preference order $\prec_i$ over the set of axioms O_i for each agent i, and a probability $prob_h$ of agent h to take a turn.

The algorithm iteratively builds an ontology R for the combined concept, initialised with O_{init}. The two agents take turns randomly following the probability distribution $(prob_h, 1 - prob_h)$. When it is her turn, agent i will choose her preferred axiom φ in O_i according to $\prec_i$, and not already entailed by the combination R. As long as φ can not be added to R without causing an inconsistency, it is replaced by one of its weakenings in $g_R(\varphi)$, i.e. a weakening of φ wrt. the current combination R. As soon as φ can be added to R without causing an inconsistency, the combination R is augmented with φ. When all the axioms of an agent have been considered or are already entailed by the current combination, this agent is finished. This iterative process continues until all agents have finished. At the end, the combination R is returned.

In the experiments, we also consider a **bounded variant** of this algorithm, where a maximum number *max_turns* of turns is added as a parameter, and where the **while**-loop exits after at most *max_turns* iterations. We now state a few formal properties of these two algorithms. It is easy to see that the returned ontology R is always consistent.

Proposition 1 (Consistency). *If O_{init} is consistent and Algorithm 1 (or its bounded variant) returns R, then R is consistent as well.*

Also, as a corollary of [25, Th. 2], we can show that the algorithm almost always terminates when using the refinement operators of [10].

Proposition 2 (Termination). *If $prob_h \notin \{0, 1\}$, then Algorithm 1 (and its bounded variant) terminates with probability 1.*

Moreover, we can formulate a sufficient condition for the combination R to be maximal in the following sense:

Proposition 3 (Maximality). *Let R be an ontology returned by Algorithm 1 (or by its bounded variant with $max\_turns \geq |O_h \cup O_m|$) and let φ be an axiom in $O_h \cup O_m$. Then $R \cup \{\varphi\}$ is either inconsistent or equivalent to R.*

We can readily use the algorithm for asymmetric concept combination of a Head concept H described by an ontology O_h with a Modifier concept M described by an ontology O_m. The result is an ontology intended to describe the target concept MH, which is the asymmetric hybridisation of the Head concept H with the Modifier concept M.

Probability $prob_h$. The asymmetry of the hybridisation must be enforced by a suitable weight given to the Head and to the Modifier, and an appropriate probability to take turns in the dialogue. In the asymmetric case, the Head agent h will be given a greater weight than the Modifier agent, agent m; it will have relatively more opportunities to insert its information into the hybridisation. One then needs to translate these weights into a probability for the Head and Modifier agents to take turns. In practice, one needs to consider the granularity of the information contained in both ontologies. At the time of the combination, an agent with a high granularity ontology, i.e. many detailed axioms, is likely to require more turns to add its information into the blend. To this end, we need a way to quantify the amount of information in an ontology. We take into account the logical axioms in a given ontology and also the inferred class hierarchy. Given an ontology O, we define the set of axioms in the *inferred class hierarchy* as follows: $\mathsf{inf}(O) = \{C \sqsubseteq D \mid C, D \in N_C \cap \mathsf{sub}(O), C \sqsubseteq_O D\}$, where $\mathsf{sub}(O)$ is the set of subconcepts appearing in O. This provides us with a useful measure to evaluate the 'amount' of information contained in an ontology O, namely defined as the quantity $|O \cup \mathsf{inf}(O)|$. Combining two ontologies O_h and O_m with weights w_h and w_m, agent h will play $w_h \cdot |O_h \cup \mathsf{inf}(O_h)| = f_h$ turns when agent m will play $w_m \cdot |O_m \cup \mathsf{inf}(O_m)| = f_m$ turns. The probability passed as a parameter to the algorithm is then $prob_h = \frac{f_h}{f_h + f_m}$.

Preferences $\prec_h$ and $\prec_m$. The preferences of the agents represent the importance of their axioms in expressing certain features of the concept at issue, for the purpose of the specific combination. We take them here as given inputs, provided by the agents, and they partially determine the 'direction' of the combination.

Initial Ontology O_{init}. The choice of the initial ontology is motivated by the goal of combining two concepts. So, when combining H and M, the initial ontology O_{init} will contain the two axioms: $\mathsf{MH} \sqsubseteq H$ and $\mathsf{MH} \sqsubseteq M$, where MH is the target hybrid concept. This is enough to bootstrap the formalisation of the requirement that the hybrid concept is an H that is also an M. Moreover, to avoid the trivial result where the resulting ontology is consistent but the hybrid concept is unsatisfiable we must also add an axiom $\mathsf{MH}(a)$ for a fresh individual name a, making sure that some MH's do exist.

5. Computational Simulations of Impossible Combinations: The FishVehicle

We illustrate how the two versions of our algorithm work in the case of an *impossible combination*. Namely, we simulate the combination of the concepts

Fish and Vehicle as it is described in Hampton's experiments [6] by means of our dialogue implementations.[3] We start with a consistent initial ontology, which will guide our weakening procedure. To provide some of the high level ontological distinctions needed for representing the input concepts, we include in our initial ontology an excerpt of the taxonomy of DOLCE, formulated in $\mathcal{ALC}$. DOLCE (i.e. Descriptive Ontology for Linguistic and Cognitive Engineering) is a cognitively oriented Foundational Ontology, which enables a fine-grained analysis of concepts. As such, it provides a number of basic ontological and cognitive distinctions required to represent and confront the common sense concepts at issue. The most general categories of DOLCE include Perdurants, Endurants, Abstracts and Qualities (see [26]). We are mostly interested in Endurants, and in particular in the distinctions related to Physical Endurants, which branch into Amount of Matter,Features and Physical Objects. Fish and Vehicle are indeed Physical Objects. Food and Fuel, for instance, which appear in our input ontologies in axioms such as "fish eats food", "vehicles need fuel", are here included in Amount of Matter. To state that "fish has slippery surface", we include Slippery Surface in Feature. We also use Qualities from DOLCE, to represent, for instance, the shape, the colour, the spatial locations of Fish and Vehicles. DOLCE Abstract and Space Region, i.e. values for spatial location qualities, are required to classify places such as Air, Ground, and Water, which we use in axioms such as "fish swims in water". Moreover, Abstracts allow for introducing conceptual spaces, to model the values of the individual qualities (e.g. colour, shape, weights) of fish and vehicles.

DOLCE serves to provide, via alignment, the ontological distinctions needed to represent and reason about the possible incompatibilities between the two concepts to be combined. Aside from DOLCE, the initial ontology contains also two additional axioms, which directly relate to the concept we want to build (as described in the section above): $\mathsf{FishVehicle} \sqsubseteq \mathsf{Fish}$ and $\mathsf{FishVehicle} \sqsubseteq \mathsf{Vehicle}$. To ensure the concept FishVehicle is not empty, we also add an instance of the concept: $\mathsf{FishVehicle(Wanda)}$.

We need then two ontologies, which represent the concepts of Fish and Vehicle respectively, before the combination can be started. These can be seen as micro-theories, little domain ontologies, modelling the two concepts involved. In our setting, they are associated with two different agents, and each axiom corresponds to a possible move in the dialogue. The specific content of the ontology of Vehicle (resp. Fish) is partly reverse-engineered using the information contained in Hampton's experiments described above (e.g. a fish *eats food*, is *autonomous* and can *swim*; a vehicle *needs fuel*, *is controlled*, and can *move*, etc.). Additional information (e.g. body parts, vehicle component, etc.) is inspired by the Leuven Database ([27]), which collects psychological, commonsense data on a feature generation task of 15 concepts (including fish and vehicle).

In order to make the two input ontologies of fish and vehicle interoperable, they are aligned to the common upper level provided by DOLCE (to achieve that, the classes of the domain ontologies are subsumed under the pertinent categories of DOLCE). DOLCE is particularly well-suited to account for some of the distinctions observed by Hampton in his experiments. E.g., it can

---

[3] See https://bitbucket.org/troquard/ontologyutils/src/master/.

capture the distinction between the agency of the Fish (modelled as a subclass of AgentivePhysicalObject) contrasting the non-agency of the Vehicle (i.e. a NonAgentivePhysicalObject), asserted indirectly through the possibility to control it.

The goal of the procedure is to build the concept of FishVehicle, which should share both features of the concept of Fish and features of the concept of Vehicle. When the algorithm starts, at each turn the agents will try to add their favourite axioms to the initial ontology. If the axiom cannot be added without causing inconsistency, it is weakened by the procedure.

We have two agents: agent_h represents the Head concept (in this case, Vehicle) and agent_m represents the Modifier (Fish). To implement the asymmetry of the combination, we do not distribute the turns equally between the two agents. At each round, the weight for agent_h to play is higher than the one for agent_m. Having the possibility to play its favourite axioms sooner, agent_h is more likely to add less weakened information to the initial ontology.

The last important aspect to consider is the preference order that we put on the axioms. As already mentioned, if an axiom is preferred and added sooner to the initial ontology, it will be less likely that it causes an inconsistency and is weakened. We consider three different preference orders. Firstly, we consider an order which enforces the strength of the ontological distinctions, i.e. the link between the ontologies of Vehicle (resp. Fish) with DOLCE. Secondly, we consider the opposite situation, i.e. where the specific axioms of Vehicle (resp. Fish) were preferred. Finally, for the domain ontologies we here follow a preference order aiming at replicating the process of instantiation as described by Hampton [7] and outlined in Section 2. In this case, agent_h preferred all the axioms containing information related to Car. In contrast, we left the Fish order as a random order.

The unbounded version of our algorithm ends, in any case, when both agents have done all their possible moves, and we obtain a maximally informative ontology R about FishVehicle. The bounded version ends after the selected number of moves, returning a consistent ontology R for FishVehicle.

6. Evaluating Asymmetric Concept Hybridisations

Distinguishing between good and bad hybridisations is neither a straightforward task nor an entirely new one. Research on it appears both in computational creativity when evaluating machine-generated combinations [28], and in cognitive psychology, where identifying human strategies and cognitive heuristics are the focus [5]. In his experiments, Hampton asked two independent judges to evaluate on a 1 to 10 scale the ‘success’ of the responses given by the subjects to an impossible combinations task [7]. However, this does not tell us much about how to effectively evaluate computationally the outcome of an impossible combination.

Lacking this kind of information, what we can measure is what kind of effects our strategies show on the output of the algorithm. We therefore consider next two parameters, namely the *asymmetry* of the combination and its *hybridity*.

Asymmetry. The asymmetry of the combination represents the relative effect of the Head concept (e.g., Vehicle) and the Modifier concept (e.g., Fish) in the result ontology R. To measure this asymmetry we exploit a ratio of preserved information from [24]: a ratio with values between 0 and 1, measuring how much information from an ontology O_i is present in another ontology R. The *ratio of preserved information* from O_i in R is

$$\mathsf{rpi}(O_i, R) = \frac{|\{\varphi \in O_i \cup \mathsf{inf}(O_i) \mid \varphi \text{ is entailed by } R\}|}{|O_i \cup \mathsf{inf}(O_i)|} \ .$$

To measure the asymmetry of the combination R, we then use the difference between the ratios of preserved information, specifically, the difference between the ratio of preserved information from O_h (the ontology of the Head) in R and the ratio of preserved information from O_m (the ontology of the Modifier) in R. The measure of *asymmetry* of R is thus the number: $\mathsf{rpi}(O_h, R) - \mathsf{rpi}(O_m, R)$.

The larger the absolute value of the asymmetry, the more one of the two concepts dominates the result ontology R. Further, a positive value indicates that the Head does dominate the Modifier in the result ontology R, whilst a negative weight instead means that the Modifier does.

Hybridity. Intuitively, a hybrid of Fish and Vehicle should share some of its features with Fish and some others with Vehicle. To formally capture this intuition, we introduce here the notion of *hybrid description*. A hybrid description of Fish and Vehicle is something like a ‘shark that needs fuel’ or ‘is made of metal and has fins’. More generally, we define the set of hybrid descriptions for a Head concept H and a Modifier M as the set of conjunctions $C_h \sqcap C_m$ such that C_h (resp. C_m)

(1) is an ascendant or descendant of H within O_h (resp. of M within O_m), and
(2) is **not** a (sub-)concept appearing in the ontology O_m describing M (resp. O_h describing H).

Formally, let O_h be the ontology defining the head concept H (e.g., $\mathsf{Vehicle}$) and O_m the ontology defining the modifier concept M (e.g., Fish). Let MH be the target hybrid concept (e.g., $\mathsf{FishVehicle}$), defined through the result ontology R. For a concept D within ontology O, define the set of ascendants and descendants (the ‘lineage’ of D in O) as $\mathsf{lin}(O, D) = \{C \in \mathsf{sub}(O) \mid C \sqsubseteq_O D \text{ or } D \sqsubseteq_O C\}$. Then, to measure the hybridity of the concept MH, we count the number of times in which $\mathsf{MH} \sqsubseteq C_h \sqcap C_m$, for $C_h \in \mathsf{lin}(O_h, H) \setminus \mathsf{sub}(O_m)$ and $C_m \in \mathsf{lin}(O_m, M) \setminus \mathsf{sub}(O_h)$, over the total number of hybrid descriptions. Notice, crucially, that we exclude the ascendants and descendants which are shared by the two concepts. Formally, with $\Lambda_H = \mathsf{lin}(O_h, H) \setminus \mathsf{sub}(O_m)$ and $\Lambda_M = \mathsf{lin}(O_m, M) \setminus \mathsf{sub}(O_h)$, the measure of *hybridity* of R is the number

$$\frac{|\{(C_m, C_h) \mid C_h \in \Lambda_H, C_m \in \Lambda_M, \text{ and } \mathsf{MH} \sqsubseteq C_h \sqcap C_m \text{ is entailed by } R\}|}{|\Lambda_H| \times |\Lambda_M|} \ .$$

We evaluated the output of our algorithms on the preference orders and parameters introduced above. We present our findings next.

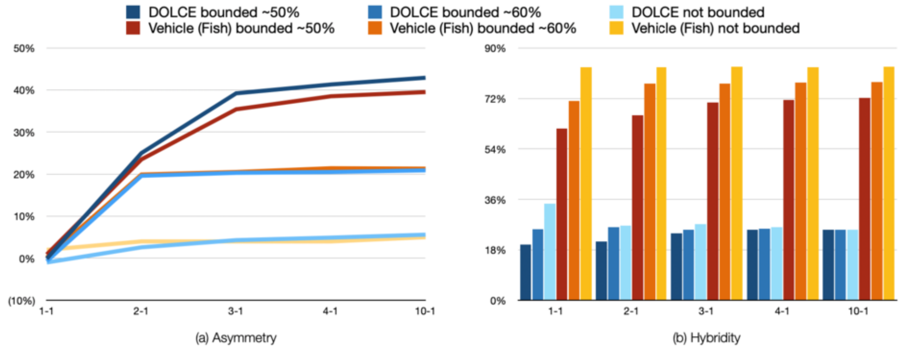


Figure 1. Asymmetry and hybridity values for the two preference orders, with varying weight of the Head. Bounded ∼50% refers to the bounded variant of Algorithm 1.

Controlling the Effects of DOLCE. In order to enforce constraints from DOLCE, agents prefer all the axioms that bridge the classes of DOLCE within our O_{init} and the classes pertaining more strictly to the ontology of Vehicle (resp. Fish). For instance, agents prefer axioms such as $\mathsf{Artefact} \sqsubseteq \mathsf{NonAgentivePhysicalObject}$. Agents also prefer the constraints imposed on the roles (Domain and Range). We expected that enforcing the link with DOLCE, and emphasising hard ontological distinctions, would have had a negative effect on the hybridity value.

The opposite strategy enforces the specific axioms for Vehicle (resp. Fish). The preference gave priority to all the axioms containing the concept Vehicle (resp. Fish) on the left or right side of the subsumption relation (e.g. $\mathsf{Vehicle} \sqsubseteq \mathsf{hasComponent.VehiclePart}$ or $\mathsf{Car} \sqsubseteq \mathsf{Vehicle}$). Enforcing first the specific information for the concepts to be combined, and only establishing the link with DOLCE at the end of the procedure, was expected to enhance the number of hybridisations, and thus increase the hybridity value.

At the same time, increasing the weight associated to a specific concept, i.e. increasing that agent's probability to play, was expected to increase the asymmetry between the two concepts.

This was confirmed by our experiments, as can be seen in Figure 1. Increasing the weight of the first player tends to increase the asymmetry value. This is particularly evident in the bounded cases. The value of hybridity was affected by the preference order, and prioritising the link with DOLCE critically decreases the number of hybridisations. The weights of the two players do not affect the hybridity values. By contrast, by setting boundaries, hybridity decreases.

Simulating Hampton's Findings. As described in Section 2, Hampton observed the tendency of people to instantiate (or specialise) general classes in order to find a solution to an impossible combination task. When instantiating the concept Vehicle into, e.g., Car, the combined concept should show some of the distinctive features of the instantiated concept. The effectiveness of an instantiation strategy should then be evaluated on the capability of the combined concept to satisfy the specific features of the instantiated concept. Similarly to the methodology of competency question employed in knowledge engineering [29], we selected to this

	Priority DOLCE	Priority Vehicle/Fish	Instantiation
FV goes on ground	4%	2%	100%
FV has brake	0	0	100%
FV has motor	0	0	100%
FV has steering wheel	0	0	100%
FV has wheel	0	0	100%
FV does not go on air or water	16%	2%	100%
FV has not wing	18%	4%	100%

Table 1. Percentage of combinations satisfying the instantiation questions, over 50 runs.

end the questions described in Table 1, which correspond to the description of the concept Car within the ontology of Vehicle.

To replicate this phenomenon within our set-up, we first include an additional axiom in our initial ontology, enforcing the FishVehicle to be also a sub-concept of Car. Then, the agent_h prefers all the axioms containing information related to Car (i.e. the concept Car occurs on the right or left hand side of a subsumption). The effectiveness of this strategy is shown in Table 1[4]. Namely, in all of 50 runs, the FishVehicle showed *all* the features associated to Car (i.e. 100%). In our tests, the weight for agent_h is set to be 3, whereas the one for agent_m is set to be 1; a probability of about 0.75 for agent_h to take a turn. In the case of the unbounded procedure, the hybridity value was high, with an average of 0.7. The 50% bounded version, at about 0.36, cuts that value of hybridity to about half. Although one might have expected such an effect given the reduced opportunity to impose 'hybrid' information, the effect is here surprisingly strong.

Another phenomenon observed by Hampton [6] is the use of alignments. According to his findings, people tend to align the fact that fish eat food with the fact that vehicles need fuel; or, the fact that both have the capacity to move. To replicate this phenomenon, the following set of axioms was then added to the initial ontology: $\mathsf{Food} \equiv \mathsf{Fuel}$; $\exists\ \mathsf{needs.Fuel} \equiv \exists\ \mathsf{eats.Food}$; $\exists\ \mathsf{swimsIn.Water} \equiv \exists\ \mathsf{goesOn.(Water} \sqcup \mathsf{Air} \sqcup \mathsf{Ground)}$; $\mathsf{Water} \equiv \mathsf{Air}$; $\mathsf{Water} \equiv \mathsf{Ground}$; $\mathsf{Ground} \equiv \mathsf{Air}$. The alignments were not, as was to be expected, consistent with the ontologies of the two players. Therefore, it was part of the weakening procedure to integrate them consistently.

We expected that introducing the alignments within our procedure would have had a positive effect on the hybridity value. This was, however, not observed within our dataset. Looking at the effects of the alignments, the main benefit observed was in terms of *feature emergence*. Introducing the alignments between Fuel and Food and between Air, Ground and Water produced in fact some mixed axioms, which were present neither in the ontology of Fish, nor in the ontology of Vehicle. Table 2 shows an example of this effect.

	Alignment	No alignment
FV eats fuel	21	0
FV swims on air or ground	12	0

Table 2. Number of emergent features over 50 runs, on a random order, unbounded procedure.

---

[4] We report here the values for the unbounded procedure, but the result is analogous for the bounded one.

7. Conclusions and Future Directions

We developed a dialogue-based algorithm for the computational generation of hybrid, sometimes considered ‘impossible’, combinations. Our method is inspired by the empirical research in psychology identifying human heuristics for combining concepts that lack any obvious similarities. To explore the dynamics involved in our dialogue games and to experimentally evaluate the human heuristics mentioned, we defined and implemented a number of measures including ratio of preserved information (rpi), hybridity, asymmetry of the combination, and impact of alignment with an upper ontology (here, DOLCE). In general, the unbounded dialogue game allows for the construction of ‘almost perfect conjunctions’ in the sense that preservation of information remains high whilst hybridity increases. This is a feature of interest more generally to ontology engineering. Further, the lack of high asymmetry in the unbounded combination can be traced back to the fact that, as Prop. 3 showed, we construct maximal combinations in this case. In contrast, the use of a bounded procedure permits to build highly asymmetrical combinations, arguably more in line with the distinction between Head and Modifier as described in cognitive psychology. As may be expected, this is obtained at the cost of a decrease in hybridity.

We also showed the flexibility of our algorithm in reproducing some of the phenomena observed in the cognitive psychology of impossible combinations, namely the use of alignments and instantiation. To simulate human concept combination in a subtler way, a more fine-grained protocol regarding evaluation of preferences, prioritisation strategies and resource-bounding should be investigated further.

References

[1] The Office, Season 5; 2009. Episode: Cafe Disco. Warren Lieberstein and Halsted Sullivan (writers) and Randall Einhorn (director).

[2] Murphy GL. The Big Book of Concepts. MIT press; 2002.

[3] Rosch E. Principles of Categorization. In: Margolis E, Laurence S, editors. Concepts: Core Readings. vol. 189. MIT press; 1999. p. 189-206.

[4] Hampton JA. Inheritance of attributes in natural concept conjunctions. Memory & Cognition. 1987;15(1):55-71.

[5] Gibbert M, Hampton JA, Estes Z, Mazursky D. The Curious Case of the Refrigerator–TV: Similarity and Hybridization. Cognitive Science. 2012;36:992-1018.

[6] Hampton JA. Emergent Attributes in Combined Concepts. In: Ward TB, Smith SM, Viad J, editors. Creative thought: An investigation of conceptual structures and processes. Washington DC: APA Press; 1997. p. 83-110.

[7] Hampton JA. Compositionality and Concepts. In: Hampton JA, Winter Y, editors. Compositionality and Concepts in Linguistics and Psychology. Cham: Springer International Publishing; 2017. p. 95-121.

[8] Wisniewski EJ. When concepts combine. Psychonomic Bulletin & Review. 1997;4:167-83.

[9] Confalonieri R, Kutz O. Blending under Deconstruction: The roles of logic, ontology, and cognition in computational concept invention. Annals of Mathematics and Artificial Intelligence. 2020;(88):479-516.

[10] Troquard N, Confalonieri R, Galliani P, Peñaloza R, Porello D, Kutz O. Repairing Ontologies via Axiom Weakening. In: Proc. of AAAI 2018; 2018. p. 1981-8.

[11] Ontañón S, Plaza E. Amalgams: A Formal Approach for Combining Multiple Case Solutions. In: Bichindaritz I, Montani S, editors. Case-Based Reasoning. Research and Development. Berlin, Heidelberg: Springer Berlin Heidelberg; 2010. p. 257-71.

[12] Besold TR, Kühnberger KU, Plaza E. Towards a computational- and algorithmic-level account of concept blending using analogies and amalgams. Connection Science. 2017;29(4):387-413.

[13] Lieto A, Pozzato GL. A description logic framework for commonsense conceptual combination integrating typicality, probabilities and cognitive heuristics. J Exp Theor Artif Intell. 2020;32(5):769-804.

[14] Eppe M, Maclean E, Confalonieri R, Kutz O, Schorlemmer M, Plaza E, et al. A computational framework for conceptual blending. Art Int. 2018;256:105-29.

[15] van den Berg L, Atencia M, Euzenat J. Agent ontology alignment repair through dynamic epistemic logic. In: Proc. of AAMAS 2020. ACM; 2020. p. 1422-30.

[16] Thorn PD, Schurz G. How Category Selection Impacts Inference Reliability: Inheritance Inference From an Ecological Perspective. Cognitive Science. 2021;45(4):e12971.

[17] Markmar AB, Gentner D. Splitting the differences: A structural alignment view of similarity. Journal of Memory and Language. 1993;32:517-35.

[18] Costello FJ, Keane MT. Efficient creativity: Constraint-guided conceptual combination. Cognitive Science. 2000;24:299-349.

[19] Jackendoff R. English Noun-Noun Compounds in Conceptual Semantics. In: The Semantics of Compounding. Cambridge University Press; 2016. p. 15–37.

[20] Righetti G, Porello D, Kutz O, Troquard N, Masolo C. Pink Panthers and Toothless Tigers: Three Problems in Classification. Manchester, Sept. 10–11: CEUR-WS, Vol-2483; 2019. Proc. of the 5th Int. Workshop on Artificial Intelligence and Cognition.

[21] Porello D, Kutz O, Righetti G, Troquard N, Galliani P, Masolo C. A Toothful of Concepts: Towards a theory of weighted concept combination. CEUR-WS, Vol-2373; 2019. Proceedings of the 32nd International Workshop on Description Logics.

[22] Baader F, Horrocks I, Lutz C, Sattler U. An Introduction to Description Logic. Cambridge University Press; 2017.

[23] van der Laag PRJ, Nienhuys-Cheng SH. Completeness and properness of refinement operators in inductive logic programming. The Journal of Logic Programming. 1998;34(3):201-25.

[24] Porello D, Troquard N, Penaloza R, Confalonieri R, Galliani P, Kutz O. Two Approaches to Ontology Aggregation Based on Axiom Weakening. In: Proc. of 27th Int. Joint Conf. on Artificial Intelligence and 23rd European Conf. on Artificial Intelligence; 2018. p. 1942-8.

[25] Confalonieri R, Galliani O Pietro Kutz, Porello D, Righetti G, Troquard N. Towards Even More Irresistible Axiom Weakening. CEUR-WS, Vol-2663; 2020. Proc. of 33rd International Workshop on Description Logics.

[26] Masolo C, Borgo S, Gangemi A, Guarino N, Oltramari A. WonderWeb Deliverable D18. CNR; 2003.

[27] De Deyne S, Verheyen S, Ameel E, Vanpaemel W, Dry MJ, Voorspoels W, et al. Exemplar by feature applicability matrices and other Dutch normative data for semantic concepts. Behavior Research Methods. 2008;40(4):1030-48.

[28] Veale T. From Conceptual Mash-ups to Badass Blends: A Robust Computational Model of Conceptual Blending. In: Computational Creativity. Springer; 2019. p. 71-89.

[29] Grüninger M, Fox MS. Methodology for the Design and Evaluation of Ontologies. IJCAI-95, Montreal; 1995. Workshop on Basic Ontological Issues in Knowledge Sharing.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210374

Debugging Classical Ontologies Using Defeasible Reasoning Tools

Simone COETZER and Katarina BRITZ
CAIR, Information Science Dept, Stellenbosch University

Abstract

A successful application of ontologies relies on representing as much accurate and relevant domain knowledge as possible, while maintaining logical consistency. As the successful implementation of a real-world ontology is likely to contain many concepts and intricate relationships between the concepts, it is necessary to follow a methodology for debugging and refining the ontology. Many ontology debugging approaches have been developed to help the knowledge engineer pinpoint the cause of logical inconsistencies and rectify them in a strategic way. We show that existing debugging approaches can lead to unintuitive results, which may lead the knowledge engineer to opt for deleting potentially crucial and nuanced knowledge. We provide a methodological and design foundation for weakening faulty axioms in a strategic way using defeasible reasoning tools. Our methodology draws from Rodler's interactive ontology debugging approach and extends this approach by creating a methodology to systematically find conflict resolution recommendations. Importantly, our goal is not to convert a classical ontology to a defeasible ontology. Rather, we use the definition of exceptionality of a concept, which is central to the semantics of defeasible description logics, and the associated algorithm to determine the extent of a concept's exceptionality (their ranking); then, starting with the statements containing the most general concepts (the least exceptional concepts) weakened versions of the original statements are constructed; this is done until all inconsistencies have been resolved.

Keywords. knowledge representation and reasoning, formal ontology, ontology debugging tools, defeasible description logic

1. Introduction

With ontologies becoming increasingly important across many industries (see for instance [1,2,3]), the need for a strategic debugging methodology is clear. Furthermore, ontologies are being applied to more abstract and complex domains such as business and law. Indeed, in the FOIS 2020 selected papers, we saw the prevalence of topics relating to these realms (see for instance [4,5,6]). Often in these domains we find that knowledge is described by referring to typical properties of a specific concept, but for specific cases the general theory can be overwritten. Arguably, humans naturally employ a method of non-monotonic reasoning when building up knowledge: facts are assumed to be correct until an exception to the facts is encountered – the Quinean web of knowledge is then slightly adjusted [7]. This highlights that not only is a *strategic* debugging methodology necessary, but that the debugging methodology should be able to deal with *logical nuances*.

Multiple debugging tools have been developed precisely so that a more strategic methodology for pinpointing the root cause faulty axioms is established (see for instance [8,9,10]). Notably, recently Schekotikin, et al. [11] created an interactive ontology debugging tool. This methodology has been instantiated in a Protégé tool, *OntoDebug*, in which the queries are methodically and iteratively posed to the user until a single diagnosis is identified, at which point the user can then make a repair.

This approach is a step in the right direction to guiding the user in the debugging process. However, in the case where what we call 'multi-level exceptions' occur this approach can sometimes lead to unintuitive results. Consider for instance the following *incoherent* ontology (i.e., some concepts are unsatisfiable, and cannot be instantiated in any model of the ontology):

$$\mathscr{O} = \left\{ \begin{array}{l} \mathsf{User} \sqsubseteq \neg\exists \mathsf{accessTo.ConfidentialInfo} \\ \mathsf{User} \sqsubseteq \exists \mathsf{accessTo.PublicInfo} \\ \mathsf{Staff} \sqsubseteq \mathsf{User} \\ \mathsf{Staff} \sqsubseteq \exists \mathsf{accessTo.ConfidentialInfo} \\ \mathsf{BlackListedStaff} \sqsubseteq \mathsf{Staff} \\ \mathsf{BlackListedStaff} \sqsubseteq \neg\exists \mathsf{accessTo.ConfidentialInfo} \end{array} \right\}$$

When the OntoDebug tool is run, the suggestion is to remove the axiom stating that Staff have access to some ConfidentialInfo. Indeed, from a classical ontology perspective, the suggestion of this repair makes sense – we have asserted that a User (the concept subsuming Staff) does not have access to some ConfidentialInfo, and we have also asserted that BlacklistedStaff (the concept subsumed by Staff) does not have access to some ConfidentialInfo.

Yet, intuitively we know that there are exceptions to the rule, and that Staff is an exception to the more general concept of User, and that BlacklistedStaff is the exception to the concept of Staff. In this case, the intuition that we would like to maintain the User-Staff-BlackListedStaff hierarchy stems from the fact that User has access to PublicInfo, a property which we would like Staff and BlackListedStaff to inherit. We know, therefore, that a more accurate repair would be to change the first and fourth axioms to read semantically as follows: User *usually* does not have access to some ConfidentialInfo; Staff *usually* do have access to some ConfidentialInfo.

We therefore propose an extension to the interactive ontology debugging methodology, which allows for recommendations to be made on how axioms can be weakened rather than deleted to further the goal of knowledge retention. Importantly, our goal is not to convert a classical ontology to a defeasible ontology – therefore we do not use defeasible reasoning support through, for example, the computation of rational closure. Rather, we use the definition of exceptionality of a concept, which is central to the semantics of defeasible description logics, and the associated algorithm to determine the extent of a concept's exceptionality (their ranking); then, starting with the statements containing the most general concepts (the least exceptional concepts) weakened versions of the original statements are constructed; this is repeated until all inconsistencies have been resolved.

Our approach differs from repair strategies that remove (parts of) axioms, possibly after computing smaller laconic or precise justifications [12]. Instead, our methodology aims to identify missing parts of axioms and add them.

The remainder of the paper is structured as follows: In Section 2 we present the necessary background to non-monotonic reasoning required for the development of our debugging strategy. Section 3 provides the necessary background on the interactive ontology debugging tool, OntoDebug, that our solution uses as a basis. Then in Section 4, we explain our extension to the interactive ontology debugging tool – this extension allows the user to view recommendations on how faulty axioms can be *weakened* rather than *deleted* through the use of defeasible DL tools. In the final section, we conclude on the contributions of our work, and what the focus of future work could be.

2. Defeasible Description Logics

The notion of defeasibility originates from non-monotonic logics. First described by McDermott and Doyle [13], the notion of non-monotonic logics were formed in contrast to monotonic or classical logics. McDermott and Doyle [13] argue that classical monotonic logics do not take into account that our human knowledge is incomplete and thus, with the addition of new facts, old facts may become invalidated or weakened.

Several non-monotonic extensions of DLs exist [14,15,16]. Britz et al. [17,18] extended the work of Kraus, Lehmann and Magidor (KLM) [19] beyond propositional logics to DLs, and their extension includes an implementation. They provide a semantic account of both preferential and rational subsumption relations based on the standard semantics of description logics. The same benefits that are obtained by using the KLM approach on propositional logic are realised when extending this approach to DLs. The KLM approach provides natural and intuitive semantics for defeasible subsumption. In the context of ontology debugging, the main benefit of using the KLM approach lies in the fact that it allows for defeasible subsumption problems to be reduced to classical entailment checking – this also has the effect that defeasibility can be introduced without increasing the computational complexity associated with classical DL reasoning tasks.

The concept language $\mathscr{L}$ of the description logic $\mathscr{ALC}$ is built according to the following rule:

$$C := \top \mid \bot \mid A \mid \neg C \mid C \sqcup C \mid C \sqcap C \mid \forall r.C \mid \exists r.C$$

Given $C, D \in \mathscr{L}$, $C \sqsubseteq D$ is called a subsumption statement, or general concept inclusion (GCI), read 'C is subsumed by D'. Defeasible subsumption, also referred to as defeasible concept inclusion, is intended as defeasible counterpart of classical subsumption. Given concepts C and D from $\mathscr{L}$ a *defeasible concept inclusion axiom* (DCI, for short) is a statement in the form $C \mathrel{\sqsubseteq\!\!\!\sim} D$.

Statements that are written in the form $C \mathrel{\sqsubseteq\!\!\!\sim} D$ should be read as 'C is *usually* subsumed by D' or 'individuals that are typical C's are also elements of D'. The symbol $\mathrel{\sqsubseteq\!\!\!\sim}$ denoting defeasible subsumption can thus be used in the same way as classical subsumption $\sqsubseteq$, the difference being that it refers to defeasible concept inclusion, and that the inclusion may be violated for exceptional individuals.

As is the case with classical subsumption $\sqsubseteq$, its defeasible counterpart $\mathrel{\underset{\sim}{\sqsubset}}$ also acts as a connective positioned between the concept language at the object level, and the meta-language (the level of entailment). The semantics of $\mathrel{\underset{\sim}{\sqsubset}}$ is defined formally w.r.t. preferential interpretations. Reasoning tasks can however be reduced to classical reasoning without affecting complexity, and has been implemented as a plugin on the Protégé platform.

The definitions below, and the algorithms to implement basic defeasible reasoning tasks, are discussed and motivated in detail in [18]. Readers who are not familiar with the theoretical foundations of the KLM approach to defeasible reasoning can focus on the intuitive explanations we provide, without engaging with the deeper mathematical foundations. We also assume familiarity with the semantics of standard description logic such as $\mathscr{ALC}$, and build on its syntax and semantics:

Definition 1. *A* ***preferential interpretation*** *is a structure* $\mathscr{P} := \langle \Delta^{\mathscr{P}}, \cdot^{\mathscr{P}}, \prec^{\mathscr{P}} \rangle$ *where* $\langle \Delta^{\mathscr{P}}, \cdot^{\mathscr{P}} \rangle$ *is a DL interpretation (which we denote by* $\mathscr{I}_{\mathscr{P}}$ *and refer to as the classical interpretation associated with* $\mathscr{P}$*), and* $\prec^{\mathscr{P}}$ *is a strict partial order on* $\Delta^{\mathscr{P}}$ *(i.e.,* $\prec^{\mathscr{P}}$ *is irreflexive, transitive and asymmetric) satisfying the smoothness condition (for every* $C \in \mathscr{L}$*, if* $C^{\mathscr{P}} \neq \emptyset$*, then* $\min_{\prec^{\mathscr{P}}}(C^{\mathscr{P}}) \neq \emptyset$*).*

Definition 2. *A defeasible subsumption relation* $\mathrel{\underset{\sim}{\sqsubset}}$ *is a* ***preferential subsumption relation*** *if it satisfies the following set of properties, called the preferential KLM properties for DLs:*

$$(Ref)\ C \mathrel{\underset{\sim}{\sqsubset}} C \qquad (LLE)\ \frac{C \equiv D,\ C \mathrel{\underset{\sim}{\sqsubset}} E}{D \mathrel{\underset{\sim}{\sqsubset}} E} \qquad (And)\ \frac{C \mathrel{\underset{\sim}{\sqsubset}} D,\ C \mathrel{\underset{\sim}{\sqsubset}} E}{C \mathrel{\underset{\sim}{\sqsubset}} D \sqcap E}$$

$$(Or)\ \frac{C \mathrel{\underset{\sim}{\sqsubset}} E,\ D \mathrel{\underset{\sim}{\sqsubset}} E}{C \sqcup D \mathrel{\underset{\sim}{\sqsubset}} E} \qquad (RW)\ \frac{C \mathrel{\underset{\sim}{\sqsubset}} D,\ D \sqsubseteq E}{C \mathrel{\underset{\sim}{\sqsubset}} E} \qquad (CM)\ \frac{C \mathrel{\underset{\sim}{\sqsubset}} D,\ C \mathrel{\underset{\sim}{\sqsubset}} E}{C \sqcap E \mathrel{\underset{\sim}{\sqsubset}} D}$$

Along with the above properties, if the relation $\mathrel{\underset{\sim}{\sqsubset}}$ also satisfies rational monotonicity (RM), then it is a *rational* subsumption relation:

$$(\mathrm{RM})\ \frac{C \mathrel{\underset{\sim}{\sqsubset}} D,\ C \not\mathrel{\underset{\sim}{\sqsubset}} \neg E}{C \sqcap E \mathrel{\underset{\sim}{\sqsubset}} D}$$

Definition 3. *Given* $C, D \in \mathscr{L}$*, a statement of the form* $C \mathrel{\underset{\sim}{\sqsubset}} D$ *is a defeasible subsumption statement. A preferential interpretation* $\mathscr{P} = \langle \Delta^{\mathscr{P}}, \bullet^{\mathscr{P}}, \prec \mathscr{P} \rangle$ ***satisfies*** *a defeasible subsumption statement* $C \mathrel{\underset{\sim}{\sqsubset}} D$*, if* $\min_{\prec \mathscr{P}}(C^{\mathscr{P}}) \subseteq D^{\mathscr{P}}$*.*

It is desirable for the defeasible entailments to adhere to rational monotonicity as it is a prerequisite for the *presumption of typicality* to hold. The presumption of typicality states that all individuals are considered to be most normal unless they are proven to be exceptional. This is central to the notion of a rational preferential ordering.

Preference orders allow individuals or objects (and, by extension, also concepts and statements) to be ordered or ranked based on their level of exceptionality relative to other individuals, concepts or statements in an ontology. In a propositional setting, this takes the form of an ordering on worlds. An object's normality or typicality is determined not by some intrinsic characteristic that the object possesses, but rather in relation to the other objects in the domain. The assumption of rationality (RM) imposes a further restriction on preference orders, namely that they are *modular*. This partitions the domain into layers that are linearly ordered.

Definition 4. *Given a set* $X, \prec \subseteq X \times X$ *is a* ***modular order*** *if it is a strict partial order, and its associated incomparability relation* $\sim$*, defined by* $x \sim y$ *if neither* $x \prec y$ *nor* $y \prec x$*, is transitive.*

Definition 5. *A* ***modular interpretation*** *is a preferential interpretation* $\mathscr{R} = \langle \Delta^{\mathscr{R}}, \bullet^{\mathscr{R}}, \prec \mathscr{R} \rangle$ *such that* $\prec^{\mathscr{R}}$ *is modular.*

Definition 6. *A statement* α *is* ***modularly entailed*** *by a defeasible knowledge base* $\mathscr{O}$*, written* $\mathscr{O} \models_{mod} \alpha$*, if every modular model of* $\mathscr{O}$ *satisfies* α*.*

However, it turns out that modular entailment represents a monotonic entailment relation, which thus reduces entailment from a knowledge base to classical reasoning. Furthermore, modular entailment is not necessarily rational. In order to obtain a non-monotonic entailment relation that is also rational, we need to look beyond a Tarskian-style consequence relation.

Our focus in this paper is not on rational entailment, but rather on the notion of *exceptionality*, a central building block in the computation of rational closure, a form of rational entailment.

Definition 7. *Let* $\mathscr{O}$ *be a defeasible knowledge base and* $C \in L$*. We say* C *is* ***exceptional*** *in* $\mathscr{O}$ *if* $\mathscr{O} \models_{mod} \top \sqsubseteq\!\!\!\!\sim \neg C$*. A DCI* $C \sqsubseteq\!\!\!\!\sim D$ *is exceptional in* $\mathscr{O}$ *if* C *is exceptional in* $\mathscr{O}$*.*

Intuitively, an exceptional concept C w.r.t. a knowledge base $\mathscr{O}$ is one to which no normal individual in the domain of $\mathscr{O}$ can belong. This definition of an exceptional concept is used in [18] to compute the *rank* of a concept. Briefly, the more exceptional a concept is, the higher is its rank. Using the defeasible counterpart of the User-Staff-BlacklistedStaff example from Section 1, the rankings would be computed as follows:

1. First, the left-hand-side concept of all defeasible statements that are non-exceptional (according to Definition 7) are given a ranking of 0. The DCIs with non-exceptional left-hand side concepts are also given a rank of 0. In this case, the concept User is assigned a rank of 0.
2. Then, a new knowledge base is created containing only the remaining exceptional statements along with the classical General Concept Inclusions (GCIs) in the knowledge base. For the left-hand side concepts of defeasible statements that are now deemed to be non-exceptional, a ranking of 1 is given to left hand side concept contained in the axiom. The DCIs with a non-exceptional left-hand side concept are also given a rank of 1. In this case, the concept Staff is assigned a rank of 1.
3. The above procedure from step 2 is repeated and with each iteration, the ranking of the left hand side concept is increased by 1. In this case, the concept BlacklistedStaff is assigned a rank of 2.
4. Once all the DCIs have been ranked, or there are no new non-exceptional concepts in the last step, if there are any concepts that remain they are given a rank of ∞.This means that the concept is, even when preferential ordering has been applied, unsatisfiable. In our example ontology, there are no further statements to assess, and so no concepts are assigned a rank of ∞.

These rankings can then be used to determine what is rationally entailed by a knowledge base:

Definition 8. *$C \sqsubsetsim D$ is in the rational closure of a knowledge base $\mathcal{O}$ if*

$$rank(C \sqcap D) < rank(C \sqcap \neg D) \text{ or } rank(C) = \infty.$$

Intuitively, this definition states that no normal object can belong to C but not to D. Such objects must be the exception in any preferential model.

3. Interactive Ontology Debugging

Basic ontology debugging focuses on finding justifications for inconsistencies in a faulty ontology. Although the basic concepts assist with fault *identification* in ontologies, an exponential number of minimal conflict sets may exist for the exceptions in an ontology. Thus, there is a need for fault *localisation* – i.e. not returning all axioms from all conflict sets, but presenting the user with only the axiom(s) that represent the root cause of the problem. In the ontology debugging community, then, it has been suggested that background knowledge, along with positive and negative test cases, should be explicitly provided as input by the user so that the test cases along with the background knowledge eliminate some of the axioms that are returned in the minimal conflict set [11].

Definition 9. *Let $\mathcal{O}$ be an ontology, and let $\mathcal{B} \subseteq \mathcal{O}$ be the* background knowledge *to $\mathcal{O}$. Then all axioms in $\mathcal{B}$ are assumed to be correct. In the context of ontology debugging, the remainder of axioms in $\mathcal{O}$ are considered potentially faulty [20].*

Background knowledge constitutes axioms that the oracle or knowledge engineer knows to be true before starting with testing. In the OntoDebug tool, the dialogue on background knowledge gets populated by the Abox statements. In the absence of Abox statements, Abox statements are auto-generated for each concept.

Positive and negative test cases are usually formulated once the knowledge engineer or oracle starts with their testing, and through the testing they uncover:

- axioms that they do not want to exist in future (negative test cases), or
- axioms that they do want to exist in future, but which were at a stage in testing not present (positive test cases).

Definition 10. ***Positive test cases** (aggregated in the set P) correspond to desired entailments of the correct (repaired) ontology, $\mathcal{O}$ along with the background knowledge $\mathcal{B}$. Each test case $p \in P$ is a set of axioms over language $\mathcal{L}$. The meaning of a positive test case $p \in P$ is that some axiom p (or the conjunction of axioms P in the case of a set of p) must be entailed by the correct $\mathcal{O}$ integrated with $\mathcal{B}$ [20].*

Definition 11. ***Negative test cases** (aggregated in the set N) represent undesired entailments of the correct (repaired) ontology $\mathcal{O}$, along with the background knowledge $\mathcal{B}$. Each test case $n \in N$ is a set of axioms over language $\mathcal{L}$. The meaning of a negative test case $n \in N$ is that some axiom n (or the conjunction of axioms N in the case of a set of N) must not be entailed by the correct $\mathcal{O}$ integrated with $\mathcal{B}$ [20].*

Once background knowledge, and positive and negative test cases are provided for the ontology, this is put together in a diagnosis problem instance (DPI) which gives the parameters in which the diagnosis should be calculated.

Definition 12. *Let $\mathcal{O}$ be an ontology (including possibly faulty axioms) and $\mathcal{B}$ be background knowledge (including correct axioms) where $\mathcal{O} \cap \mathcal{B} = \emptyset$, and let $\mathcal{O}^*$ denote the (unknown) intended ontology. Moreover, let P and N be sets of axioms where each $p \in P$ must and each $n \in N$ must not be entailed by $\mathcal{O}^* \cup \mathcal{B}$, respectively. Then, the tuple $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$ is called a* ***diagnosis problem instance (DPI)*** *[11].*

Definition 13. *Let $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$ be a DPI. Then, a set of axioms $\mathcal{D} \subseteq \mathcal{O}$ is a* ***diagnosis*** *if and only if both of the following conditions hold:*

1. $(\mathcal{O} \setminus \mathcal{D}) \cup P \cup \mathcal{B}$ *is consistent (coherent if required)*
2. $(\mathcal{O} \setminus \mathcal{D}) \cup P \cup \mathcal{B} \not\models n$ *for all* $n \in N$

A diagnosis $\mathcal{D}$ is minimal iff there is no $\mathcal{D}' \subset \mathcal{D}$ such that $\mathcal{D}'$ is a diagnosis [11].

If background knowledge, positive and negative test cases are incorporated when diagnoses are determined, this will limit the number of potentially faulty axioms that are output as explicit instructions are given as to which entailments and axioms can be deemed correct or incorrect [20]. Rodler's suggestion is to automate the process of finding test cases by developing an algorithm which, targeting the most likely diagnoses first, iteratively asks the knowledge engineer (in this case, someone who is referred to as the 'oracle' – someone who has full knowledge of a given domain) whether certain axioms should or should not be entailed.

Definition 14. *Let* $\boldsymbol{Ax}$ *be a set of axioms and* $ans : \boldsymbol{Ax} \rightarrow P \cup N$ *a function which assigns axioms in* $\boldsymbol{Ax}$ *to either the positive or negative test cases. Then, we call ans an* ***oracle*** *w.r.t. the intended ontology $\mathcal{O}^*$, iff for each $ax \in \boldsymbol{Ax}$ both the following conditions hold:*

1. $ans(ax) = P \rightarrow \mathcal{O}^* \cup \mathcal{B} \models ax$
2. $ans(ax) = N \rightarrow \mathcal{O}^* \cup \mathcal{B} \not\models ax$

[11].

A query is a set of axioms which, once the knowledge engineer/ oracle provides an answer as to whether the entailments should hold or not, sufficient information is obtained such that at least one diagnosis can be eliminated.

Definition 15. *Let $\langle \mathcal{O}, \mathcal{B}, P, N \rangle$ be a DPI, $\mathcal{D}$ be a set of diagnoses for this DPI, and Q be a set of axioms. Then we call Q a* ***query*** *for $\mathcal{D}$ iff, for any classification Q^P_{ans}, Q^N_{ans} of the axioms in Q of a domain expert oracle ans, at least one diagnosis in $\mathcal{D}$ is no longer a diagnosis for the new DPI $\langle \mathcal{O}, \mathcal{B}, P \cup Q^P_{ans}, N \cup Q^N_{ans} \rangle$ [11].*

The knowledge engineer's answers to these queries are added to the list of test cases. The process of posing queries to the knowledge engineer, and feeding through the knowledge engineer's answer, and recomputing the new diagnoses is performed until only minimal number faulty axioms remain for each DPI.

4. Defeasible Reasoning Support for Interactive Ontology Debugging

As illustrated in the Introduction, multi-level exceptions can lead to unintuitive results and loss of information when axioms are removed while following Rodler's interactive debugging methodology.

We propose that Rodler's [20] original interactive ontology debugging methodology be followed until an unintuitive result is obtained. If the interactive ontology debugging methodology is followed, and we get to an unintuitive suggestion for an axiom to repair, the following methodology is followed:

1. **Isolate the issue:** Create a separate sub-ontology, $\mathscr{O}'$ containing the axiom listed for repair, along with axioms that, from the minimal conflict sets, lead to this axiom being identified as a potentially faulty axiom.
2. **Determine a candidate axiom to weaken, and a candidate weakening concept with which to weaken the candidate axiom:** To determine this, the ranking algorithm is used on the above ontology $\mathscr{O}'$: central to the ranking formula is the notion of exceptionality.
   (a) Ranking of 0 – least exceptional: First we identify the concepts with a rank of 0. In this case this would be User. Then, the statements with a rank of 0 are identified.
   (b) Ranking of 1 – concepts that are exceptional w.r.t. level 0 statements: $\mathscr{O}''$ now contains only the remaining exceptional statements after the axioms that now have an associated ranking have been removed.

Per the ranking algorithm, concepts now have the following ranking:

World order/ rank	Concept
0	User
1	Staff

Table 1. First iteration concept ranking output.

The axioms are then ranked to correspond to the ranking of the respective LHS concept. Therefore, it follows that the axioms have the following ranking:

World order/ rank	Axiom
0	$\mathsf{User} \mathrel{\underset{\sim}{\sqsubset}} \neg\exists \mathsf{accessTo.ConfidentialInfo}$
1	$\mathsf{Staff} \mathrel{\underset{\sim}{\sqsubset}} \mathsf{User}$
1	$\mathsf{Staff} \mathrel{\underset{\sim}{\sqsubset}} \exists \mathsf{accessTo.ConfidentialInfo}$

Table 2. First iteration axiom ranking output.

It should be noted that even though in a minimal conflict set there may be concepts that are ranked at a level higher than 1, only concepts (and axioms) at levels 0 and 1 will be used in the next step. Furthermore, it is only ever necessary to work on these two levels to systematically resolve multi-level exceptions as the conflicts preceding the next level would have been solved already.

3. **Weaken the relevant axiom:** Next, the postulate of Cautious Monotonicity is applied to weaken the axiom at level 0. As referenced in Definition 2:

$$\text{(CM)}\ \frac{C \mathrel{\sqsubset\!\!\!\sim} D,\ C \mathrel{\sqsubset\!\!\!\sim} E}{C \sqcap E \mathrel{\sqsubset\!\!\!\sim} D}$$

   The weakened result we would like to get to has a form similar to that of the axiom below the line: $C \sqcap E \mathrel{\sqsubset\!\!\!\sim} D$. In our case, the weakened result would be $\mathsf{User} \sqcap \neg\mathsf{Staff} \mathrel{\sqsubset\!\!\!\sim} \neg\exists\mathsf{accessTo.ConfidentialInfo}$. Thus we find that in the postulate of Cautious Monotonicity, C can represent User, D can represent $\neg\exists\mathsf{accessTo.ConfidentialInfo}$ and E can represent $\neg\mathsf{Staff}$:

$$\text{(CM)}\ \frac{\mathsf{User} \mathrel{\sqsubset\!\!\!\sim} \neg\exists\mathsf{accessTo.ConfidentialInfo},\ \mathsf{User} \mathrel{\sqsubset\!\!\!\sim} \neg\mathsf{Staff}}{\mathsf{User} \sqcap \neg\mathsf{Staff} \mathrel{\sqsubset\!\!\!\sim} \neg\exists\mathsf{accessTo.ConfidentialInfo}}$$

   The rule that is extrapolated here is thus that when using Cautious Monotonicity to apply weakening to an axiom at level 0, use the axiom as is for the first premise (top left axiom) in the postulate; for the second premise (top right axiom), use the subsumed (left hand) concept at level 0 subsumed by the negation of a concept at level 1; the resultant conclusion (bottom axiom) is then the axiom showing the weakened result. This step is mandated by Lemma 1 below.

4. **Choose to accept or reject solution:** The classical counterpart of the defeasible axiom obtained by applying Cautious Monotonicity is what is then displayed to the knowledge engineer as a repair recommendation, and the can choose to accept or reject.

5. **Repeat until done:** This process is repeated until all inconsistencies have been resolved.

Lemma 1. *Let $\mathcal{O}$ be a defeasible knowledge base, and let C and E be concepts with* $rank(C) = 0$ *and* $rank(E) = 1$. *It then follows that* $C \mathrel{\sqsubset\!\!\!\sim} \neg E$ *is in the rational closure of* $\mathcal{O}$.

Proof. Since $rank(C) = 0$, it follows that either $rank(C \sqcap E) = 0$ or $rank(C \sqcap \neg E) = 0$. But since $rank(E) = 1$, $rank(C \sqcap E) \geq 1$. Therefore, $rank(C \sqcap \neg E) = 0$, and hence $rank(C \sqcap \neg E) < rank(C \sqcap E)$. It follows from Definition 8 that $C \sqcap \neg E$ is in the rational closure of $\mathcal{O}$. □

This lemma shows that the Cautious Monotonicity (CM) rule is applicable to an axiom with subsumed (lefthand) concept C at rank 0 by left strengthening with the negation of any concept at rank 1. The result can be generalised to concepts with rank greater than 1, but the case considering an axiom at rank 0 and left strengthening concepts at rank 1 is the most interesting because throughout the execution of the suggested methodology, it is only concepts at rank 0 and rank 1 that are considered.

This extension to OntoDebug is visually depicted in Figure 1. The extension is shown in green, while the original OntoDebug methodology is in blue.

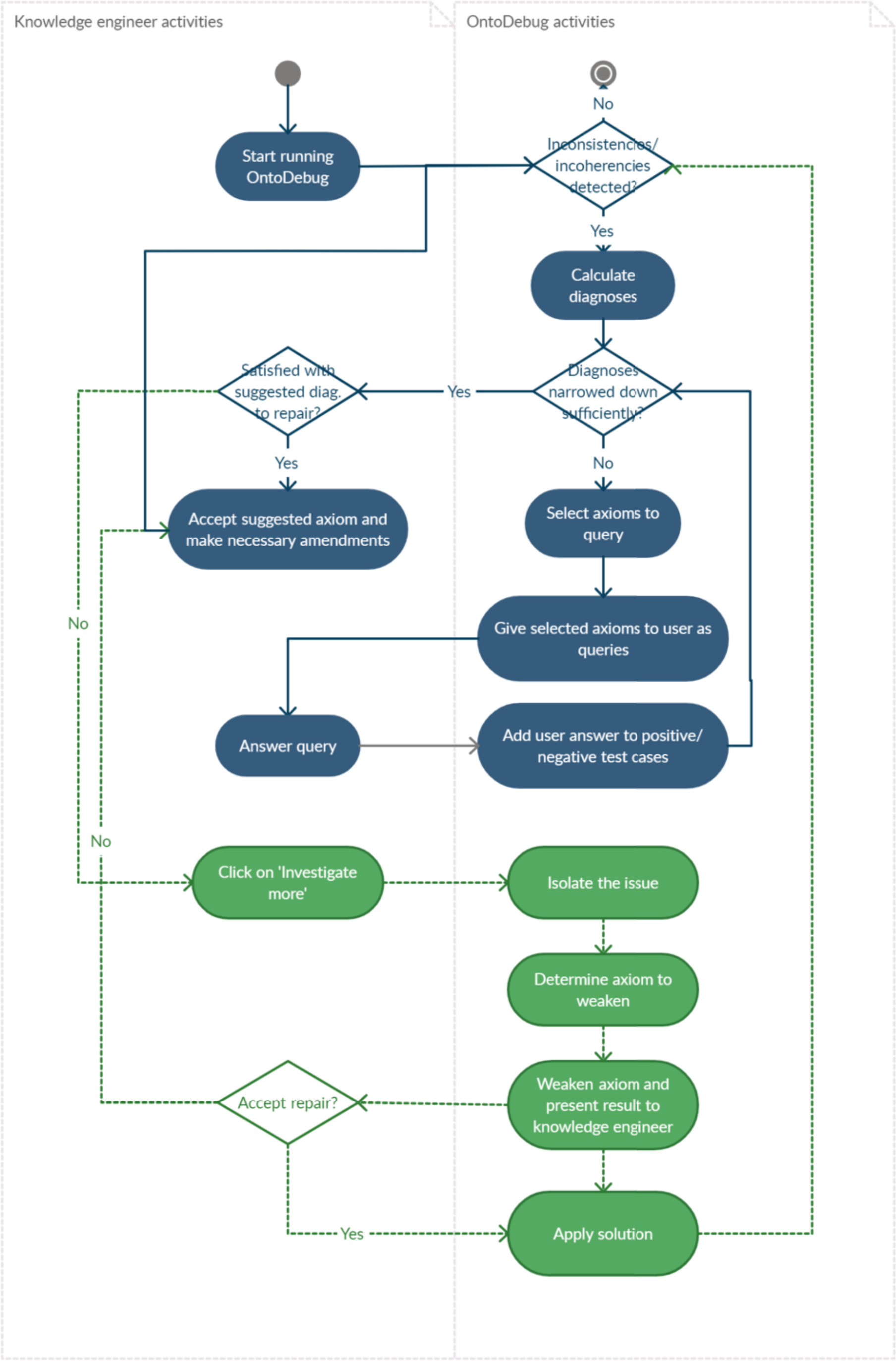


Figure 1. OntoDebug extension methodology

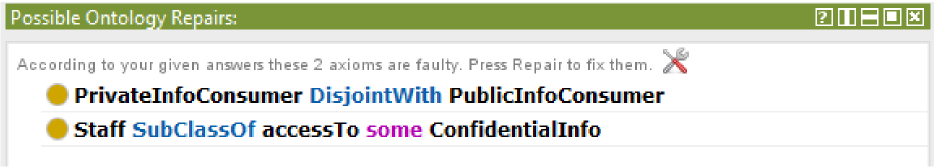
4.1. Applying Axiomatic Weakening to a More Complex Example

To illustrate how the above methodology will be carried out in practice, and work together with the standard interactive ontology debugging framework, consider the following ontology:

$$\mathscr{O} = \left\{ \begin{array}{l} \mathsf{Staff} \sqsubseteq \mathsf{User} \\ \mathsf{User} \sqsubseteq \exists \mathsf{accessTo.PublicInfo} \\ \mathsf{User} \sqsubseteq \neg \exists \mathsf{accessTo.ConfidentialInfo} \\ \mathsf{Staff} \sqsubseteq \exists \mathsf{accessTo.ConfidentialInfo} \\ \top \sqcap \exists \mathsf{accessTo.PublicInfo} \sqsubseteq \mathsf{PublicInfoConsumer} \\ \top \sqcap \exists \mathsf{accessTo.ConfidentialInfo} \sqsubseteq \mathsf{PrivateInfoConsumer} \\ \mathsf{PrivateInfoConsumer} \sqsubseteq \neg \mathsf{PublicInfoConsumer} \\ \mathsf{ConfidentialInfo} \sqsubseteq \neg \mathsf{PublicInfo} \end{array} \right\}$$

In this example, an entangled inconsistency is present: Staff is an unsatisfiable concept for two reasons: firstly, Staff is unsatisfiable because it is asserted that Staff have $\mathsf{accessTo.ConfidentialInfo}$, yet at the same time, because Staff is subsumed by User, it is also inferred that Staff do not have $\mathsf{accessTo.ConfidentialInfo}$. Secondly, Staff is an unsatisfiable concept because it is inferred that Staff is subsumed by $\mathsf{PrivateInfoConsumer}$ because Staff have $\mathsf{accessTo.ConfidentialInfo}$ and anything that has $\mathsf{accessTo.ConfidentialInfo}$ is considered a $\mathsf{PrivateInfoConsumer}$. Yet, Staff is also subsumed by User, and it is inferred that User is subsumed by $\mathsf{PublicInfoConsumer}$ because a User has $\mathsf{accessTo.PublicInfo}$ and anything that has $\mathsf{accessTo.PublicInfo}$ is considered a $\mathsf{PublicInfoConsumer}$. The incoherence occurs because the concepts $\mathsf{PrivateInfoConsumer}$ and $\mathsf{PublicInfoConsumer}$ are asserted as being disjoint, yet the concept of Staff has been identified as both a $\mathsf{PrivateInfoConsumer}$ and a $\mathsf{PublicInfoConsumer}$.

When following through with the standard OntoDebug methodology, we see that for the above example, two axioms are suggested as in need of repair, due to two minimal conflict sets being involved in causing the incoherence:



According to your given answers these 2 axioms are faulty. Press Repair to fix them.

● **PrivateInfoConsumer DisjointWith PublicInfoConsumer**

● **Staff SubClassOf accessTo some ConfidentialInfo**

Figure 2. Running entangled concepts through OntoDebug returns separate axioms for repair.

Thus, each repair axiom can be examined individually. Simply removing the axiom asserting disjointness between $\mathsf{PrivateInfoConsumer}$ and $\mathsf{PublicInfoConsumer}$ solves the first axiom to be repaired – this is also a good example of where it may at times be necessary to simply remove an axiom, rather than attempting to weaken it, as it does make logical sense that someone who is a $\mathsf{PublicInfoConsumer}$ can also be a $\mathsf{PrivateInfoConsumer}$.

It does not however make sense to remove or alter the second axiom. In this case, the knowledge engineer can choose to investigate further with the extended debugging methodology to identify relevant axioms that could be weakened rather than deleted. If the knowledge engineer chooses to investigate further in this case, the same steps listed at the beginning of Section 4 will be followed, leading to the result that the recommendation for weakening is $\mathsf{User} \sqcap \neg\mathsf{Staff} \precsim \neg\exists\mathsf{accessTo.ConfidentialInfo}$. Of course, if further levels of exceptionality are present (e.g. the axioms with $\mathsf{BlackListedStaff}$ on the LHS), in the next iteration these are picked up on as needing repair, and the same methodology as at the beginning of Section 4 is again followed.

4.2. Axiomatic Weakening as an Ontology Design Pattern

Axiomatic weakening can work both as part of a model-based diagnosis or heuristic approach to ontology debugging. Thus far, we have worked with model-based diagnosis for debugging ontologies. The heuristic approach to debugging tries to find common patterns of faulty ontology modelling and presents suggestions for repairs based on this [21]. The benefit of using the heuristic approach is that, especially with large ontologies, computation of repairs is more efficient as minimal conflict sets do not need to be computed for each inconsistency before returning a result.

Gangemi and Presutti [22] describe an ontology design pattern as a "modelling solution to solve a recurrent ontology design problem". In this case the recurrent ontology design problem is unintuitive exceptionality due to axioms that are stated too strongly. Abstracting away from the User-Staff-$\mathsf{BlacklistedStaff}$ example that we have been using up until now, we may define this kind of exception as follows:

Definition 16. *An **exceptionality pattern** is a recurrent ontology design problem that occurs when, in an ontology $\mathcal{O}$, a concept, H which intuitively must be subsumed by the parent concept, G, causes an inconsistency due to having a relationship r with another concept, I, which is in direct opposition to the relationship that the parent concept G has with the other concept, I.*

That is:

$$\mathcal{O} = \left\{ \begin{array}{l} G \sqsubseteq I \\ H \sqsubseteq G \\ H \sqsubseteq \neg I \end{array} \right\}$$

and intuitively the knowledge engineer would like to still maintain that all of the above axioms are true.

The modelling solution to this recurrent ontology design problem is to weaken the axiom with the most general concept (with the lowest rank) on the left hand side by left-strengthening the most general concept on the left hand side by adding a conjunction with the exceptional concept, as follows: $G \sqcap \neg H \sqsubseteq I$.

5. Conclusion and Future Work

Formal ontologies serve as knowledge representation formalisms over which reasoning tasks can occur. In a vast array of domains, they can be used to formalise knowledge so that axioms are machine-readable and can be reasoned over thus sourcing new knowledge and identifying domain inconsistencies. The success of ontologies thus depends on (1) knowledge retention, (2) without introducing undue logical inconsistencies.

As ontologies are being used in more domains, and especially in domains such as business or legal where there are often exceptions to the rules, the axiomatic intricacy (variety) increases, meaning that inconsistencies *arise more unexpectedly* and evade understanding of how they came about. As inconsistencies arise more often, faster and more frequently evade understanding, the human ability to find adequate solutions for these inconsistencies becomes impaired.

In the same way that Rodler et al. [20,21,11] could motivate the necessity of an interactive ontology debugging methodology by arguing that without it, valuable axioms are often deleted thus leading to a loss of knowledge, our extension can also be motivated: without a strategy showing *how* axioms could be weakened rather than deleted, valuable knowledge may be lost.

For each diagnosis, our extension suggests a way to fix the inconsistency / incoherency by weakening rather than deleting a relevant axiom in the minimal conflict set of that diagnosis. From the point where the knowledge engineer decides to investigate a particular diagnosis returned by OntoDebug in more detail, this is done by:

1. Isolating the issue by pulling through only the selected minimal conflict set (our methodology provides recommendations on which minimal conflict sets would be more apt to address first, though the onus still lies with the knowledge engineer);
2. Determining a candidate axiom to weaken and a candidate concept with which to weaken it by obtaining the ranking of concepts within the minimal conflict set.
3. Weakening the relevant axiom by applying Cautious Monotonicity.

The weakened axioms are returned to the knowledge engineer and they choose to accept or reject the solutions. The full OntoDebug methodology, together with our extension, is followed until all inconsistencies have been resolved, and the ontology is no longer incoherent. We have also shown that axiomatic weakening can be used as an Ontology Design Pattern as part of a heuristic approach to ontology debugging.

We have created a design artifact in the form of a methodology and design plans to suggest how, through the use of defeasible reasoning tools, suggestions of axiomatic weakening could be systematically presented to the user. Our extension enables the usage of a debugging methodology that applies the principle of minimal change in a more nuanced way, thus serving the ultimate goal of knowledge retention in an ontology. This is the main contribution of our work along with the contribution of unearthing rich areas for investigation at the intersection between the defeasible DL and debugging communities.

Future work could focus on using the existing design artifact as a blueprint for an implemented Protégé plug-in, as an extension to OntoDebug. Certain algorithms that play a significant role in the development of this extension have already been implemented: Meyer et al. [23] have, for instance, created the Defeasible Inference Platform (DIP) Protégé plug-in. This plug-in has the ability to rank concepts appearing in defeasible axioms. Furthermore, interactive ontology debugging has been implemented in the OntoDebug Protégé plug-in. Implementation would thus rely on seamlessly merging the existing algorithms, and the design artifact produced by our work will guide the developer in this process. Once implemented, future work could also study the extent to which the effectuated repairs mimic the human non-monotonic reasoning process. Ultimately, this would lead to more robust ontology repairs.

References

[1] Quamar A, Lei C, Miller D, Ozcan F, Kreulen J, Moore RJ, et al. An ontology-based conversation system for knowledge bases. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data; 2020. p. 361-76.

[2] Ziaimatin H, Nili A, Barros A. Reducing consumer uncertainty: Towards an ontology for geospatial user-centric metadata. ISPRS International Journal of Geo-Information. 2020;9(8):488.

[3] Zheng Y, Tetik M, Törmä S, Peltokorpi A, Seppänen O. A Shared Ontology for Logistics Information Management in the Construction Industry. In: ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction. vol. 37. IAARC Publications; 2020. p. 1278-85.

[4] Ismail HO, Shafie M. A Commonsense Theory of Secrets. In: Formal Ontology in Information Systems. IOS Press; 2020. p. 77-91.

[5] Griffo C, Almeida JPA, Guizzardi G. Legal Theories and Judicial Decision-Making: An Ontological Analysis. In: Formal Ontology in Information Systems. IOS Press; 2020. p. 63-76.

[6] Biccheri L, Ferrario R, Porello D. Needs and Intentionality. In: Formal Ontology in Information Systems. IOS Press; 2020. p. 125-39.

[7] Roth D. Learning to reason: The non-monotonic case. In: IJCAI; 1995. p. 1178-84.

[8] Schlobach S, Huang Z, Cornet R, Harmelen F. Debugging incoherent terminologies. Journal of automated reasoning. 2007;39:317-49.

[9] Kalyanpur A, Parsia B, Sirin E, Cuenca-Grau B. Repairing unsatisfiable concepts in OWL ontologies. In: European Semantic Web Conference. Springer; 2006. p. 170-84.

[10] Friedrich G, Shchekotykhin K. A general diagnosis method for ontologies. In: International Semantic Web Conference. Springer; 2005. p. 232-46.

[11] Schekotihin K, Rodler P, Schmid W. OntoDebug: interactive ontology debugging plug-in for Protégé. In: International Symposium on Foundations of Information and Knowledge Systems. Springer; 2018. p. 340-59.

[12] Horridge M, Parsia B, Sattler U. Laconic and Precise Justifications in OWL. In: Sheth A, Staab S, Dean M, Paolucci M, Maynard D, Finin T, et al., editors. The Semantic Web - ISWC 2008. Springer Berlin Heidelberg; 2008. p. 323-38.

[13] McDermott D, Doyle J. Non-Monotonic Logic I. Revision. Massachusetts Institute of Technology – Artificial Intelligence Laboratory; 1979.

[14] Knorr M, Hitzler P, Maier F. Reconciling OWL and non-monotonic rules for the Semantic Web. ECAI 2012. 2012.

[15] Giordano L, Gliozzi V, Olivetti N, Pozzato GL. A non-monotonic description logic for reasoning about typicality. Artificial Intelligence. 2013;195:165-202.

[16] Varzinczak I. A note on a description logic of concept and role typicality for defeasible reasoning over ontologies. Logica Universalis. 2018;12(3-4):297-325.

[17] Britz K, Casini G, Meyer T, Varzinczak I. A KLM perspective on defeasible reasoning for description logics. In: Description Logic, Theory Combination, and All That. vol. 11560 of LNCS. Springer; 2019. p. 147-73.

[18] Britz K, Casini G, Meyer T, Moodley K, Sattler U, Varzinczak I. Principles of KLM-style defeasible description logics. ACM Transactions on Computational Logic (TOCL). 2021;22(1):1-46.

[19] Kraus S, Lehmann D, Magidor M. Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence. 1990;44:167-207.

[20] Rodler P. Interactive Debugging of Knowledge Bases [Ph.D. thesis]. Alpen-Adria University Klagenfurt; 2015.

[21] Rodler P, Jannach D, Schekotihin K, Fleiss P. Are query-based ontology debuggers really helping knowledge engineers? Knowledge-Based Systems. 2019;179:92-107.

[22] Gangemi A, Presutti V. Ontology design patterns. In: Handbook on ontologies. Springer; 2009. p. 221-43.

[23] Casini G, Meyer T, Moodley K, Sattler U, Varzinczak I. Introducing defeasibility into OWL ontologies. In: International Semantic Web Conference. Springer; 2015. p. 409-26.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210375

An Infrastructure for Collaborative Ontology Development

Lessons Learned from Developing the Financial Industry Business Ontology (FIBO)

Dean ALLEMANG[a,d], Pawel GARBACZ[b,d,e], Przemysław GRĄDZKI[d,e],
Elisa KENDALL[c] and Robert TRYPUZ[b,d,e,1]

[a] *Working Ontologist LLC*
[b] *The John Paul II Catholic University of Lublin*
[c] *Thematix Partners LLC*
[d] *EDM Council*
[e] *MakoLab SA*

Abstract. Collaborative development of a shared or standardized ontology presents unique issues in workflow, version control, testing, and quality control. These challenges are similar to challenges faced in large-scale collaborative software development. We have taken this idea as the basis of a collaborative ontology development platform based on familiar software tools, including Continuous Integration platforms, version control systems, testing platforms, and review workflows.

We have implemented these using open-source versions of each of these tools, and packaged them into a full-service collaborative platform for collaborative ontology development. This platform has been used in the development of FIBO, the Financial Industry Business Ontology, an ongoing collaborative effort that has been developing and maintaining a set of ontologies for over a decade.

The platform is open-source and is being used in other projects beyond FIBO. We hope to continue this trend and improve the state of practice of collaborative ontology design in many more industries.

Keywords. ontology development tooling, continuous integration, hygiene test, collaborative ontology development, FIBO

Introduction

The development of a standard is inherently a collaborative process, regardless of the nature of the standard or the domain to which it applies. Good standards require input from a wide variety of subject matter experts from multiple organizations, working together in a highly distributed environment. The tools and process for developing a standard have to support collaboration smoothly and transparently. In this paper we focus on the situation in which one of the goals of the standard is to publish a shared data model or shared

---

[1] Corresponding Author: Robert Trypuz; E-mail:fibo@edmcouncil.org

reference data in the form of a formal ontology. This situation poses particular challenges for collaboration process and infrastructure, and is the focus of this paper. We draw on our experience in developing FIBO, the Finance Industry Business Ontology, to understand the challenges that face such a standards development effort, and outline the design of the infrastructure we have used for several years to manage those challenges.

FIBO evolved out of concerns that arose during the 2008 financial crisis among individuals who worked together in data governance and management. The most pressing issue at the time was the need for a *shared*, common vocabulary, focused on financial contracts and related concepts, that could be used for analysis and regulatory reporting purposes. From that time, FIBO has been sponsored and hosted by the Enterprise Data Management Council (EDM Council)[2], a global association for data management professionals, initially focused on financial services that has since expanded to other domains. Some selected modules of FIBO have been standardized by the Object Management Group (OMG)[3], and a new baseline standard is in work. As of the latest release[4], the production subset of FIBO includes roughly 1750 classes, 750 relationships and attributes, and over 14000 individuals (nominals, reference data and examples). The development version, which ranges in maturity from “almost releasable” to “really, really rough”, is more than 40 percent larger.

There are currently three FIBO primary content development teams working in parallel on different but related topics. In order to coordinate continuous integration of new and revised material, facilitate collaboration across topic teams, and ensure continuous quality improvement, leadership and process teams were put in place several years ago. One of the products of their work is a development framework created to automate aspects of ontology “unit-level testing", to guarantee a minimum level of quality. The individual tests are not necessarily novel. Many of them have been derived from, or inspired by, earlier work on ontology evaluation such as Chimaera [1], OntoClean [2], and OOPS! [3]. What is new, however, is a portable, open-source infrastructure that automatically runs these tests as an integral part of the ontology integration and publication process. This framework is designed for either a single development environment, or cross-organizationally, for example, for FIBO, and as planned for other ontology standards efforts at the EDM Council, OMG, and the Industrial Ontology Foundry (IOF)[5]. To date, in addition to its use for FIBO, the framework has been successfully deployed at Rensselaer Polytechnic Institute (RPI)’s Tetherless World Constellation[6], for use on several projects.

In this paper we present the infrastructure described above. The approach follows well-established principles and leverages tools commonly used in software engineering, treating *ontologies as source code*. Section 1 describes the FIBO development process, which is focused on use cases developed within working groups. Section 2 covers our approach to ontologies as managed source code, and how the infrastructure supports the development effort, with examples at various steps in the process. Finally, section 3 compares our approach with related attempts at supporting collaborative ontology development.

---

[2] See:https://edmcouncil.org

[3] See: https://www.omg.org

[4] See: https://github.com/edmcouncil/fibo/releases/tag/master_2021Q1

[5] See: http://www.industrialontologies.org

[6] See:https://tw.rpi.edu

1. Ontology Development Approach

The original motivation for FIBO was the failure of financial institutions and regulatory agencies to clearly exchange and integrate data about financial contracts and their counterparties, as demonstrated by the industry's failure to roll up risk with respect to those contracts. The initial FIBO use case was to provide an industry glossary that financial institutions and other market participants can use to meet regulatory requirements such as Dodd-Frank[7] in the U.S. and the MiFID II[8] framework in the EU for regulating financial markets. That use case was extended to cover additional requirements for data governance, data management, and enterprise glossaries mandated in the EU by the Basel Committee on Banking Supervision (BCBS) for risk data aggregation and reporting (BCBS 239[9]). Over the last few years, we have refined our approach as recommended in [4] to create instrument- or topic-specific use cases that add incremental value, resulting in significant progress by each of the working groups. The use cases include several usage scenarios and a number of competency questions per scenario, which are used to test the efficacy of the ontology as the work progresses.

The FIBO effort is organized into working groups, each consisting of at least one ontologist and some number of subject matter experts, which meet weekly to (1) review the use cases, (2) find areas in the ontologies where gaps remain, (3) refine and extend the ontologies to address those gaps and other issues raised by users, and (4) develop examples that answer the competency questions based on the revisions to the ontologies. Given an issue, use case, or partial use case, such as one scenario, the development process is roughly as follows:

1. In the context of a working group teleconference, review the existing ontology to determine what aspects of the ontology can be used to answer the question(s)
2. Identify the specific gap(s) and raise an issue to address the gap
3. Identify any missing concepts and work together to develop definitions and other annotations for those concepts and any important relationships based on a combination of appropriate resources (online financial dictionaries, offline financial dictionaries, ISO and other financial standards, etc.) and record our findings, discussion, and references in our minutes in the working group wiki
4. Create a branch in GitHub for the issue
5. Identify the ontology(ies) that need to be revised, where in the class hierarchy the concept(s) belong, and, importantly, whether or not there are existing patterns we can leverage in order to integrate the material
6. Integrate the new content into the relevant ontology(ies), reusing existing classes and properties as much as possible, and extending them as needed
7. Run at least one reasoner and perform SPARQL queries to ensure that the semantics seem reasonable and that the ontology(ies) remain logically consistent
8. Check the changes into GitHub and push them to a remote branch so that other members of the working group can review the results, automatically invoking the RDF serializer described below that ensures consistent serialization of the resulting RDF/XML via a custom Git hook

---

[7] See: https://www.govinfo.gov/content/pkg/PLAW-111publ203/html/PLAW-111publ203.htm

[8] See: https://www.esma.europa.eu/policy-rules/mifid-ii-and-mifir/

[9] See: https://www.bis.org/publ/bcbs239.pdf

9. Create example individuals (or update existing individuals) and test whether or not the competency question(s) can now be answered by the ontology (as appropriate), and check-in any examples that might be used as guidance for FIBO users
10. Once the working group members are comfortable with the revisions, perform a pull request in GitHub to get broader review, which automatically kicks off the infrastructure presented below; address any issues uncovered as a consequence
11. Once the pull request passes all of the stages in the publication cycle, at least two qualified reviewers must sign off (currently active members of at least one of the working groups plus other process team members have this privilege)
12. Finally, one of the process team will merge the pull request after it has been approved.

We iterate through steps 6-9, as needed, depending on the complexity of the issue and until we reach consensus on the resulting ontologies. Additional information regarding the methodology, minimal criteria for metadata and ontology content, and unit-level hygiene testing is outlined in our ontology guide[10].

Note that the development steps outlined above do not describe aspects of our methodology with respect to pairwise, pattern-driven development, as presented in [5]. A 'pair programming' approach that is increasingly pattern driven is applied regularly by the FIBO content teams for ontology revisions. Neither does it cover our test and integration process, which includes build out of example content, use of multiple reasoners and rule engines to ensure consistency and correctness, or validation of results against competency questions. The focus presented herein is on the automated infrastructure that is used to support the development process.

2. Ontology as Source Code

The long-term success of the process outlined in the previous section hinges on the ability to manage incremental change in the ontology (ontologies) over both short and long development cycles. This dynamic is not unique to ontology development, and in fact is quite familiar from collaborative software development methods, in which *an ontology is managed as a piece of program source code*. In our approach, we take this idea literally, and treat ontology components as source code, and organize the development using tools and techniques familiar from software engineering. In particular, we focus on four familiar areas of software development: Modularity, Version Control, Continuous Integration (CI), and Testing.

2.1. Modularity and Maturity Levels

Just as is the case with any large software project, components of a large ontology like FIBO have different governance requirements. These include ownership, speed of update, and dependencies. Just as is the case for conventional software, *modularity* is a powerful tool for managing governance. As an example from the point of view of the FIBO audience, modularization allows us to express the maturity level of different parts of the ontology,

---

[10] See: https://github.com/edmcouncil/fibo/blob/master/ONTOLOGY_GUIDE.md

allowing the community to understand which FIBO ontologies can be used "as they are" and which ones are still under development and likely to be more volatile.

FIBO is structured as a collection of relatively small ontologies; currently, there are about two hundred ontologies. Each ontology has its own namespace and is recorded (as source code) as a single OWL-compliant file, serialized as RDF/XML. Furthermore, these ontologies are organized into a hierarchical structure that is reflected in a file folder structure stored on GitHub[11].

At the top level, FIBO currently has ten modules called *domains* (represented in the file system as high-level directories). Within these domain areas are one or two levels of *subdomains*. In the smallest subdomains (bottom level directories in GitHub), are the two hundred ontology files. Thus, any ontology has a unique place in the domain/module hierarchy.

For example, for the Business Entities domain and the Legal Entities module, we have the following structure:

- **(FIBO domain)** Business Entities
  - **(FIBO module)** Corporations
  - **(FIBO module)** Functional Entities
  - **(FIBO module)** Government Entities
  - **(FIBO module)** Legal Entities
    - **(FIBO ontology)** Corporate Bodies Ontology
    - **(FIBO ontology)** Formal Business Organizations Ontology
    - **(FIBO ontology)** Legal Entity Identifier (LEI) Entities Ontology
    - **(FIBO ontology)** Legal Persons Ontology
  - **(FIBO module)** Ownership and Control
  - **(FIBO module)** Partnerships
  - **(FIBO module)** Private Limited Companies
  - **(FIBO module)** Sole Proprietorships
  - **(FIBO module)** Trusts

Each ontology in FIBO is described by one of three maturity levels[12]: *release*, *provisional*, or *informative*. Ontologies marked as *release* are considered to be stable and mature, and ready for use by stakeholders. Ontologies marked as *provisional* are considered to be under development, so they are less stable, and one can expect changes occurring in their content more frequently. Ontologies marked as *informative* have been deprecated, but are still included for informational purposes because some provisional concept references them.

2.2. Collaborative Version Control

One of the advantages of viewing ontologies as source code is that we can take advantage of decades of experience with managing collaboration in code production. The state of the art in software version control is embodied in a service called GitHub, which has become the default infrastructure for source code collaboration.

---

[11] See: https://github.com/edmcouncil/fibo

[12] Maturity levels in FIBO are assigned only to FIBO ontologies (so not to FIBO domains or modules).

GitHub's operation is based on the idea that it is possible to compare two versions of the same file and display the differences in a simple form, as shown in Figure 1. This works very well for program code, since a typical change is done using a text editor or an IDE that make changes directly to the code; any changes show up as simple differences from one version to the next. This means that if two contributors make changes to the same file, the changes from each of them can be detected, and then displayed, processed or even automatically merged.

```
-   <owl:Class rdf:about="&fibo-ind-fx-fx;FxSpotVolatility">
+   <owl:Class rdf:about="&fibo-ind-fx-fx;ExchangeRateVolatility">
        <rdfs:subClassOf rdf:resource="&fibo-ind-ind-ind;Volatility"/>
        <rdfs:subClassOf>
            <owl:Restriction>
-               <owl:onProperty rdf:resource="&fibo-ind-ind-ind;isVolatilityOf"/>
-               <owl:allValuesFrom rdf:resource="&fibo-fnd-acc-cur;ExchangeRate"/>
+               <owl:onProperty rdf:resource="&fibo-fnd-utl-alx;hasArgument"/>
+               <owl:someValuesFrom rdf:resource="&fibo-ind-fx-fx;ExchangeRateStructure"/>
            </owl:Restriction>
        </rdfs:subClassOf>
-       <rdfs:label>FX spot volatility</rdfs:label>
-       <skos:definition>measure of exchange rate fluctuation</skos:definition>
-       <fibo-fnd-utl-av:explanatoryNote>Mathematically, volatility is the annualized standard deviation of the daily changes in the exchange rat
+       <rdfs:label>exchange rate volatility</rdfs:label>
+       <skos:definition>statistical measure of the rate of change in the rate at which one currency can be exchanged for another</skos:definitic
+       <fibo-fnd-utl-av:usageNote>Volatility is modeled here using a structured collection, comprised of a series of individual exchange rates (
+   </owl:Class>
```

Figure 1. GitHub 'diff' shows the changes between two snapshots in unified diff format.

GitHub falls short in a situation in which code is not directly edited as text files, as is often the case when editing ontologies. Typically, a contributor to an ontology will want to view and edit the ontology with a user interface tuned specifically to ontology development. The actual text files that serve as the 'golden copy' of the ontology are the output from such tools.

The RDF standard includes a number of serialization options (e.g., Turtle [6] and RDF/XML [7]); each of these specifies a syntax for writing down triples in a text file. But the relationship between triples and a file is not one-to-one; any set of triples can be written down in a wide variety of ways. This is not an unusual situation; even conventional programming languages are agnostic about things like variable names, the order in which variables and subroutines are declared, etc. The difference is that for most programming languages, a human decides these things when they edit the text file; when an ontology is edited through a graphical user interface, these decisions are made by the ontology management application. Each of these many tools [13] makes different serialization decisions. This means that two versions of the same file that do not differ at all could be saved in files that are vastly different - thwarting the basis on which GitHub works.

There are three basic approaches to this problem:

- Standardization on a single tool. If this tool is consistent in the way it serializes triples, then similar ontology versions will be written in similar files, and GitHub can work appropriately. Most modern ontology editors satisfy this condition.
- Use a tool with version control built in. An example of such a system (working with RDF triples) is MOBI [14].

---

[13] https://www.w3.org/2001/sw/wiki/Tools

[14] https://mobi.inovexcorp.com/

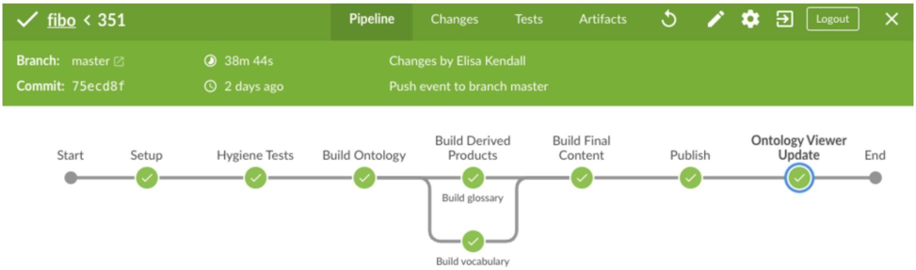
- Post-process files with a stable serializer. A hybrid approach is to allow each contributor to use whatever tool they like, but process each ontology file after it is written and before it is committed to version control. This process must preserve the content of the file (same triples in as out), but serialize it in a consistent way.

Each of these approaches has its advantages and disadvantages. Using a single tool has the advantage that contributors can collaborate easily, but has the disadvantage that it does not allow contributors to use whatever tool they choose. Using a tool with version control in it simplifies the process quite a lot, but also limits tool choices. These approaches are sometimes appropriate in an enterprise setting, where there are many motivations for controlling software use. Using a stable post-processor allows contributors to use whatever tool they like (and even encourages the development of new tools for specific uses). A disadvantage of this approach is that it requires extra infrastructure for each collaborator; they must install the serializer before they can contribute. This approach is more appropriate in a wide collaboration setting, where different collaborators come from different organizations. Not surprisingly, this latter approach is the approach that FIBO has taken. The EDM Council has provided an open-source serializer[15]. This allows ordinary text comparison tools to operate on the OWL files that represent the FIBO ontologies, and, in turn, allows FIBO developers to follow any workflow based on GitHub. In the case of FIBO, we have integrated GitHub with a continuous integration platform (see section 2.3 below) that runs a number of services and tests over each committed change.

2.3. Continuous Integration

FIBO uses a popular continuous integration (CI) platform called Jenkins; but any of several platforms perform similar functions. The job of Jenkins is to coordinate actions that will be automatically taken whenever a change is committed to GitHub.

When a change to is committed, it triggers a "chain of actions" (which is called a "pipeline"), in Jenkins, as shown in Figure 2.



Branch: master

Commit: 75ecd8f

38m 44s

2 days ago

Changes by Elisa Kendall

Push event to branch master

Figure 2. FIBO's continuous integration and continuous delivery pipeline system.

The sequence of stages in the pipeline is as follows:

- "Setup" prepares the infrastructure for the next stages,
- "Hygiene Tests" checks whether FIBO follows all the principles from the FIBO ontology guide

---

[15] See: https://github.com/edmcouncil/fibo/blob/master/CONTRIBUTING.md#fibo-serialization-tools

- “Build Ontology” is responsible for creating ontology files in different serializations,
- “Build Derived Products” creates FIBO derived products such as SKOS version of FIBO or FIBO glossary (a tabular version of FIBO containing labels and definitions, see below)
- “Build Final Content” combines the results of the building of all the products,
- “Publish” places files on the webserver,
- “Ontology Viewer Update” sends an “update” message to the Ontology Viewer

The many services used by Jenkins to manage FIBO updates are included in a software module called the ontology-publisher[16]. The ontology-publisher is deployed as a docker image, built on the basis of the Dockerfile[17] and made available on the public EDMC Docker Hub account [18]. The workhorse of this image is the “publish.sh” script[19], which enables the execution of all steps described in the caption of figure 2. The “ontology-publisher” uses the following components: Python libraries RDFLib[20] and PyLD[21], EDMC’s rdf-toolkit[22], Apache Jena[23] and other. It is built in a modular way, which allows for the flexible addition of new steps of the process and their easy integration into the Jenkins pipeline.

2.4. Testing

In conventional software development, testing is a large component of any development activity. FIBO uses Jenkins to perform a wide range of automated tests.

FIBO is developed by a heterogeneous community. In order to ensure consistency in contributions from a variety of community members, the FIBO team has developed a set of *hygiene tests* that are run automatically each time a FIBO change is committed to GitHub. FIBO organized hygiene tests into three categories:

1. errors
2. production errors
3. warnings

All three categories of tests run against the full FIBO ontology with the proposed change. If a test fails, Jenkins will take an action depending on the category of the test. If a *warning* test fails, a warning is issued, and the developer and the FIBO team are made aware of the transgression. In this case, ontology processing continues. If an *error* test fails, then processing stops, and no more steps are taken. The change is considered a fatal error, and it cannot be accepted. In the case of a *production error*, the same test is run over the *production* version of FIBO and then separately over the *development* version of FIBO; a failure in the production version of FIBO is treated as fatal (just like a "vanilla" error), whereas a failure in the development version of FIBO is treated as a warning. If no fatal error occurs, the change moves on to a review phase.

---

[16] https://github.com/edmcouncil/ontology-publisher

[17] https://github.com/edmcouncil/ontology-publisher/blob/master/Dockerfile

[18] https://hub.docker.com/repository/docker/edmcouncil/ontology-publisher

[19] https://github.com/edmcouncil/ontology-publisher/blob/master/publisher/publish.sh

[20] https://rdflib.dev/

[21] https://github.com/digitalbazaar/pyld

[22] https://github.com/edmcouncil/rdf-toolkit

[23] https://jena.apache.org/

The hygiene tests are implemented as SPARQL queries that are called by Jenkins using the Jena ARQ processor.[24] An example of such test can be found in Listing 1.

Once a commit has passed the tests, it is eligible to become a pull request. This is a concept that is common in git-based software development, whereby a contributor proposes a change to the ontology, and others approve it and merge it into the main, published branch. In the case of FIBO, a contributor proposing a pull request asserts that they have read and understood the FIBO workflow, including all the tests listed above, and signs the Developer Certificate of Origin (DCO) certifying that he/she has the right to submit the proposed change under the MIT license. The Continuous Integration system validates that these tests have passed.

```
# banner Definitions shouldn't be circular – this finds direct circularities therein.

SELECT DISTINCT ?error ?definition ?label
WHERE
{
  ?s rdfs:label ?label .
  ?s skos:definition ?definition .
  FILTER NOT EXISTS {?s a owl:NamedIndividual} .
  FILTER (REGEX(?definition, "\\W"+?label+"\\W"))
  FILTER (CONTAINS(str(?s), "edmcouncil"))

  BIND(concat("PRODERROR: Definition of ",str(?s)," is immediately circular ") AS ?error)
}
```

Listing 1: Hygiene test example

Once the pull request has been made, FIBO has a board of reviewers, two of which must approve the change, at which point (after a minimum review period has passed), it is merged into the main branch. This assures that changes make it in to FIBO in a timely fashion after undergoing peer review. This sort of workflow is typical of large, open-source code projects, in which the code undergoes some sort of unit and integration tests before it is accepted into the main branch. In the case of ontologies, the process is the same, but the tests are tailored to ontology development in OWL.

As of the last run our hygiene test finds zero errors or warnings in the production subset of FIBO and around 1000 warnings in the development subset. The latter number, although significant, is constantly decreasing over time.

2.5. Ontology Derived Products

The canonical version of FIBO is rendered in the OWL 2 DL language. Use of combinations of exact qualified cardinalities, intersections, and existential quantification to differentiate financial instruments from similar variants requires greater expressivity than is available in OWL 2 RL. For example, a fixed-float interest rate swap is a swap that has exactly two swap legs, one of which is a fixed-rate debt instrument and the other is a floating-rate debt instrument, as shown in Figure 3.

---

[24] All tests can be found on https://github.com/edmcouncil/fibo/tree/master/etc/testing/hygiene.

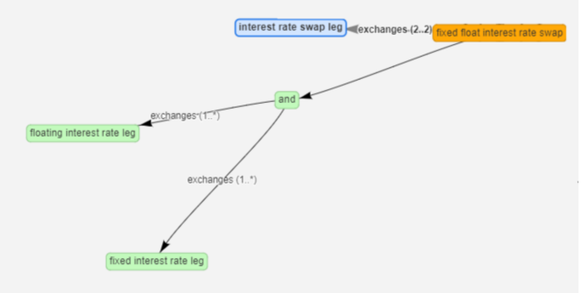


Figure 3. fixed-float interest rate swap expressed in OWL.

On the other hand, many data management applications don't require this level of detail; they really just want to know that a fixed-float interest rate swap has a fixed-rate leg and a floating-rate leg, more like the simpler diagram shown in Figure 4.

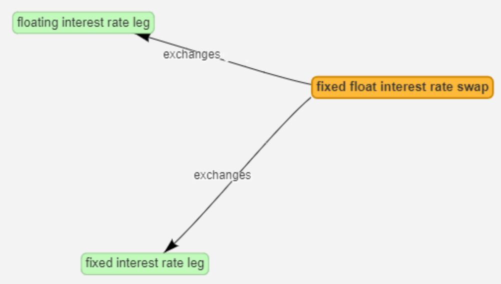


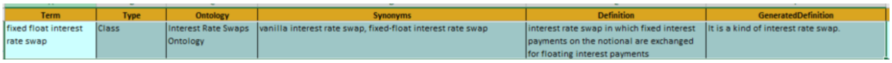
Figure 4. Fixed-float interest rate swap, in simplified form.

For this reason, FIBO is published in alternate forms that are faithfully generated from FIBO, but that include less information than the prescriptive model. For example, the structure in Figure 4 is automatically derived from the logical ontology in Figure 3; it isn't as detailed, but approximates the information. Specifically, we provide two derivative version of FIBO:

1. FIBO as a SKOS vocabulary
2. FIBO as a data dictionary

SKOS FIBO vocabulary expresses all FIBO classes as individuals in the spirit of SKOS. These individuals are related by the respective object properties. This is the structure that is shown in Figure 4.

In many cases, data managers are accustomed to seeing data elements in a spreadsheet, with names, synonyms and definitions in a tabular form. This typically expresses much less information than is available in an OWL ontology, but is amenable for human review, and for use in glossary applications. Figure 5 shows an example of this sort of display.



Term	Type	Ontology	Synonyms	Definition	GeneratedDefinition
fixed float interest rate swap	Class	Interest Rate Swaps Ontology	vanilla interest rate swap, fixed-float interest rate swap	interest rate swap in which fixed interest payments on the notional are exchanged for floating interest payments	It is a kind of interest rate swap.

Figure 5. Data dictionary entry for the fixed-float interest rate swap.

FIBO data dictionary is a single table that includes all FIBO classes, properties, and describes the basic metadata features of each one.

2.6. Ontology Viewer

Finally, in order to enhance the collaborative development of FIBO, we provide a web application to browse through its contents. The Ontology Viewer is a REST API-based web application written in Java that allows accessing the FIBO structure and its content in a user-friendly way. It is an open-source community project hosted by EDMC[25].

The Ontology Viewer provides an interactive experience that reflects all of the features we have described here: the modularity of FIBO is reflected in the navigation functionality of the viewer. The maturity of each ontology is shown. For components of the ontologies (classes and properties), they are uniformly documented with labels, synonyms and logical relationships. These are expressed in words (describing the logical relationships) as well as diagrams. Every mention of a resource in FIBO is a hyperlink that leads to more information about that entity. In addition to the human-readable display, the full URI of each resource can be copied with a single click, for use by developers when composing SPARQL queries.

The Ontology Viewer has some unique advanced features; in addition to a full-text search, it also includes a time machine. Every proposed pull request is registered in the Ontology Viewer; a reviewer can select which version of FIBO is to be viewed (including new, proposed versions as well as old, released versions). This feature is invaluable for reviewers who want to understand the ramifications of a proposed change, or to research how FIBO has changed over the years.

Ontology Viewer always presents up-to-date content; it is kept current as part of the continuous integration logic described in section 2.3. The FIBO viewer is available to the public at https://spec.edmcouncil.org/fibo/ontology.

3. Related Work

Open-source tools for collaborative ontology modeling, such as WebProtégé[8], are widely used in the knowledge representation community for specifying ontologies in the Web Ontology Language (OWL)[9]. There are fewer common processes and downstream collaboration environments, however, for development involving multiple, independently evolving ontologies, that must be kept in sync and that ensure the resulting ontologies meet the quality levels necessary for publication of an international standard.

As far as we are aware there is no publicly available framework for continuous ontology development with the same level of support for distributed, social development as the infrastructure described above. There are however a couple of web applications that provide some of the capabilities we offer – in some cases surpassing ours in terms of flexibility or scope[26].

There are a number of frameworks available today for collaborative ontology development. Most of these did not exist, however, when our work on the FIBO

---

[25] See https://github.com/edmcouncil/onto-viewer

[26] We will not discuss ontology editors here although some of them, like VocBench [10] or WebProtege [8], do provide support for collaborative ontology development.

infrastructure was initiated. The Ontology Development Tookit (ODK)[27] was developed in parallel with our work to assist the OBO Foundry community[28] in standardizing their collaborative ontology development workflows using GitHub. It supports methods of extracting various subsets of an ontology, including some axiom stripping and relaxation of axioms which we do not do in order to produce versions of a baseline ontology that can interoperate with other ontologies for certain purposes. It is possible that we will take advantage of some of their insights for FIBO users over time. ODK does not, however, automatically publish revisions covering the alternative representations, including the 'flattened' SKOS Vocabulary and data dictionary that the FIBO user community requires[29] OBO Foundry community members and others also use the ROBOT[30] semantic diff tool as a part of their process. While such a tool is quite useful, and some FIBO users also use ROBOT as part of their workflow, ROBOT does not serialize an ontology to support the file-oriented diff process that the RDF Toolkit provides, which has proven to be essential for visual comparison in GitHub for FIBO users. Another such framework is OnToology, a web-based tool designed to automate part of the ontology development process in a collaborative environment, *i.e.,* in GitHub. When one registers an ontology with OnToology, the tool will monitor for changes and upon each (committed) change it will generate a new pull request that contains: (i) the ontology documentation (with several proposals for diagram representation), its evaluation, and publication of the ontology in the user's repository[11]. VoCol is an integrated environment to support version-controlled vocabulary development. It supports a round-trip model of vocabulary development: modeling, population, and testing. To facilitate modeling VoCol allows users to formulate queries which represent competency questions for testing the expressivity and applicability of a vocabulary. To support testing, it provides the automatic detection of "bad smells" in the vocabulary design by employing SPARQL patterns. For modeling purposes, VoCol integrates a number of techniques facilitating conceptual work: automatically generated documentation and visualizations provide different views on the vocabulary as well as an evolution timeline supporting traceability[12].

There also exist various tools that automatically check the level of compliance of the ontology development against a set of logical and conceptual requirements. OOPS! is a web app that scans ontologies looking for potential pitfalls that could lead to modelling errors.[31] OOPS! is intended to be used by ontology developers during the ontology validation activity, particularly during the diagnosis task. OOPS! currently catalogs 41 checks, of which 33 are automated. Some of these automated checks overlap with the FIBO checks: P01-P03, P08, P32, P34, and P35.[13][32]. RDFUnit is a debugging framework that can run automatically generated (based on a schema) and manually generated test cases against an RDF – either inputted directly or via an endpoint[33]. The test case definition language is SPARQL, which proved to be convenient to directly query for identifying violations. For rapid test case instantiation, a pattern-based SPARQL-template

---

[27] See https://github.com/INCATools/ontology-development-kit

[28] See https://www.obofoundry.org

[29] As of this writing, we have not had the resources to test whether or not ODK supports the level of analysis over the tens of ontologies in the imports closure of some changes as a part of the testing and publication process that the FIBO infrastructure does.

[30] See http://robot.obolibrary.org/

[31] http://oops.linkeddata.es/

[32] Note that OnToology uses OOPS! for its evaluation service.

[33] https://aksw.org/Projects/RDFUnit.html

engine, running over a library of common patterns is supported where variables can be easily bound into patterns.[14]. RDFUnit thus corresponds our hygiene test component, in fact it provides more flexible functionality due to the automatic test generation. OntoSeer is a recent Protege plugin that provides automatic recommendations while ontology development [15]. These recommendations concern, among other things, IRIs for classes and properties, subsumption hierarchy structure, and recommendations based on the repository of ontology design patterns (http://ontologydesignpatterns.org/). We may consider adapting it for use with the patterns we use in FIBO in the future.

Finally, there is an emerging stream of research focused on ontology evolution, including visualization of changes in ontology design. ChImp, a Protege plugin, is an instance of this approach. It visualizes the diff between the current and the previous state of the ontology, its impact on logical consistency, and the customized metrics of on how the diff changed such aspects as the property to class ratio or the annotation richness[16].

4. Future Work

The number and nature of the SPARQL queries we run has increased considerably over the last 18 months, but more tests can be added. FIBO has evolved to be increasingly pattern oriented, as mentioned above, with recent focus on situational patterns such as ownership and control. The patterns are complicated, and it would be helpful to add tests that look for issues in their application, potentially leveraging the approach taken by OntoSeer. The same is true for other patterns, such as those involving time, payment, dividend, and other schedules, and new patterns that emerge in time. Our hope is for the framework to be widely used, so organizing the tests to allow other implementations to pick and choose which ones they care about and ignore others would be useful. We also want to make it easy for others to contribute new tests and ultimately catalog them in such a way that makes the tests searchable. Finally, we plan to integrate feedback from other implementations to evolve the framework to facilitate usage outside of FIBO.

5. Conclusion

Our experience managing FIBO, a collaborative, standardized ontology development effort, has given us some insight into how to manage such projects. The effort is daunting, and we don't claim to have got every decision right over the decade of FIBO development and maintenance; but we have learned from our mistakes, and have incorporated them into an open ontology development environment. Along the way, we realized that many of the challenges we face are common to any collaborative software effort, and so we have drawn on the experience of hundreds of open software projects to adapt their best practices to the unique requirements of ontology development.

But we also found that some challenges are unique to ontology development. Unlike most software development, many phases of ontology development are done using graphical tools, so that the ontology files themselves are not directly edited by the developers, This places special requirements on the infrastructure to enable collaboration. Ontology testing is possible, but unlike most software, there isn't a notion of "running" an ontology; you can draw inferences, and run queries over it, but there isn't an "input/output" relationship that specifies the desired behavior.

Despite these differences, we have found the parallel between code management and ontology management to be very productive, and that the workflows (e.g., commits and pull requests) and tools (version control, continuous integration) from software management do in fact apply well to ontology development.

It is our hope that we can interest more ontology development projects in using and contributing to this effort, providing much needed support across many communities and industries.

Acknowledgments

Many people have contributed FIBO and to the development, deployment, and maintenance of the infrastructure over the last decade. We would like to thank Mike Atkin and Mike Bennett for the dedication and hard work that launched FIBO, including the work they did to bring countless skeptical bankers, vendors, and consultants together to create the content at the heart of the ontologies. We would also like to recognize Dennis Wisnosky for his vision, establishing the original structure and teams that enabled scalable development of the FIBO content, and for developing the overall Build-Test-Maintain-Deploy process that led to the creation of the infrastructure. We also want to thank Anthony Coates and Omar Khan for their work on the RDF Serializer, and Jacobus Geluk and Pete Rivett for their contributions to the infrastructure.

References

[1] McGuinness DL, Fikes R, Rice J, Wilder S. The Chimaera Ontology Environment. In: AAAI/IAAI. AAAI Press / The MIT Press; 2000. p. 1123-4. Available from: http://dblp.uni-trier.de/db/conf/aaai/aaai2000.html#McGuinnessFRW00.

[2] Guarino N, Welty C. A Formal Ontology of Properties. In: Dieng R, Corby O, editors. Knowledge Engineering and Knowledge Management Methods, Models, and Tools. Springer, Berlin, Heidelberg; 2000. p. 97-112. Available from: https://www.researchgate.net/publication/2362508_A_Formal_Ontology_of_Properties.

[3] Poveda-Villalón M, Gómez-Pérez A, Suárez-Figueroa MC. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. International Journal on Semantic Web and Information Systems (IJSWIS). 2014;10(2):7-34.

[4] Kendall EF, McGuinness DL. Ontology Engineering. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers; 2019. Available from: https://doi.org/10.2200/S00834ED1V01Y201802WBE018.

[5] Shimizu C, Hirt Q, Hitzler P. MODL: A Modular Ontology Design Library. In: Janowicz K, Krisnadhi AA, Villalón MP, Hammar K, Shimizu C, editors. Proceedings of the 10th Workshop on Ontology Design and Patterns (WOP 2019) co-located with 18th International Semantic Web Conference (ISWC 2019); 2019. p. 47-58.

[6] Carothers G, Prud'hommeaux E. RDF 1.1 Turtle. W3C; 2014. Available from: http://www.w3.org/TR/2014/REC-turtle-20140225/.

[7] Schreiber G, Gandon F. RDF 1.1 XML Syntax. W3C; 2014. Available from: http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/.

[8] Horridge M, Gonçalves RS, Nyulas CI, Tudorache T, Musen MA. WebProtégé: A Cloud-Based Ontology Editor. In: Companion Proceedings of The 2019 World Wide Web Conference. New York, NY, USA: Association for Computing Machinery; 2019. p. 686–689.

[9] Bao J, Kendall EF, McGuinness DL, Patel-Schneider PF. OWL 2 Web Ontology Language Quick Reference Guide (Second Edition). W3C; 2012. Available from: https://www.w3.org/TR/owl2-quick-reference/.

[10] Stellato A, Fiorelli M, Turbati A, Lorenzetti T, Gemert W, Dechandon D, et al. VocBench 3: A collaborative Semantic Web editor for ontologies, thesauri and lexicons. Semantic Web. 2020 05;11:1-27.
[11] Alobaid A, Garijo D, Poveda-Villalón M, Santana-Perez I, Fernández-Izquierdo A, Corcho O. Automating ontology engineering support activities with OnToology. Journal of Web Semantics. 2019;57:100472.
[12] Halilaj L, Petersen N, Grangel-González I, Lange C, Auer S, Coskun G, et al. Vocol: An integrated environment to support version-controlled vocabulary development. In: European Knowledge Acquisition Workshop. Springer; 2016. p. 303-19.
[13] Gómez-Pérez A. Did You Validate Your Ontology? OOPS! The Semantic Web: ESWC 2012 Satellite Events: ESWC 2012 Satellite Events, Heraklion, Crete, Greece, May 27-31, 2012 Revised Selected Papers. 2015;7540:402-7.
[14] Dimou A, Kontokostas D, Freudenberg M, Verborgh R, Leghmann J, Mannens E, et al. Test-driven Assessment of [R2] RML Mappings to Improve Dataset Quality. In: Proceedings of the 14th International Semantic Web Conference: Posters and Demos; 2015. p. 747-58.
[15] Bhattacharyya P, Mutharaju R. OntoSeer: A Tool to Ease the Ontology Development Process. In: 8th ACM IKDD CODS and 26th COMAD. ACM; 2021. p. 428-8.
[16] Pernischova R, Serbak M, Dell'Aglio D, Bernstein A. ChImp: Visualizing Ontology Changes and their Impact in Protégé. In: Proceedings of the Fifth International Workshop on Visualization and Interaction for Ontologies and Linked Data co-located with the 19th International Semantic Web Conference (ISWC 2020). Ceur – Workshop Proceedings; 2020. p. 47-60.

III. Domain Ontology

This page intentionally left blank

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210377

NAct: The Nutrition & Activity Ontology for Healthy Living

Dorothea TSATSOU [a,1], Elena LALAMA [b], Saskia L. WILSON-BARNES [c], Kathryn HART [c], Véronique CORNELISSEN [d], Roselien BUYS [d], Ioannis PAGKALOS [e], Sofia BALULA DIAS [f], Kosmas DIMITROPOULOS [a], and Petros DARAS [a]

[a] *Information Technologies Institute, Centre for Research & Technology Hellas, Thessaloniki, Greece*
[b] *Department of Endocrinology and Metabolic Diseases, Charité Universitätsmedizin, Berlin, Germany*
[c] *School of Biosciences and Medicine, Faculty of Health and Medical Sciences, University of Surrey, Guildford, UK*
[d] *Department of Rehabilitation Sciences, Katholieke Universiteit Leuven, Leuven, Belgium*
[e] *Department of Nutritional Sciences and Dietetics, International Hellenic University, Thessaloniki, Greece*
[f] *Faculdade de Motricidade Humana, Universidade de Lisboa, Lisbon, Portugal*

Abstract. This paper presents the NAct (Nutrition & Activity) Ontology, designed to drive personalised nutritional and physical activity recommendations and effectively support healthy living, through a reasoning-based AI decision support system. NAct coalesces nutritional, medical, behavioural and lifestyle indicators with potential dietary and physical activity directives. The paper presents the first version of the ontology, including its co-design and engineering methodology, along with usage examples in supporting healthy nutritional and physical activity choices. Lastly, the plan for future improvements and extensions is discussed.

Keywords. ontology, health, healthy living, nutrition, physical activity, user modelling, recommendation, personalisation

1. Introduction

Nutrition research is a fast-moving multidisciplinary field, which combines the expertise of different professionals across different disciplines. A limitation for this research field is that it is not dependent on one variable, but on many, and to analyse this in a practical and ethical way represents a significant issue. Randomized controlled trials are the gold standard on which many dietary recommendations are predominantly based upon

---

[1] Corresponding Author: Dorothea Tsatsou, Information Technologies Institute, Centre for Research & Technology Hellas, 6th km Charilaou-Thermi road, 57001, Thermi, Thessaloniki, Greece; E-mail: dorothea@iti.gr.

(e.g., [1], [2], [3]). However, the principal limitation of these is that they are not personalised to an individual user.

Nowadays, through the use of artificial intelligence (AI) we can support an individual remotely and less invasively, through healthy lifestyle recommendations [4], for the general population, while also potentially improve the self-management of non-communicable diseases such as obesity, diabetes and cardiovascular disease [5]. To this end, knowledge based systems that rely on expert-verified knowledge enable advanced personalization of healthy lifestyle directives to each individual while at the same time adhering to consolidated and ethical guidelines of different fields of nutrition research.

To this end, this paper presents the NAct ontology, engineered based on evidence-based expert knowledge of different professionals in the nutrition, activity and health fields. Previously developed expert systems suggest the alteration of one variable for an individual's lifestyle. Whereas, NAct ontology and the knowledge-based system that employs it as the backbone for intelligent personalized decision making, aims to fill the gap by adopting a holistic approach. This approach pertains to the adoption of semantic entities and rules that connect each subject's implicit and explicit nutritional and well-being goals, and these goals with the situational condition of the subject and standardized European nutritional and well-being directives.

The structure of the document is as follows. Section 2 provides a brief overview of related work, focusing on the distinct lack of relevant ontologies and comparing NAct with the two most relevant approaches. Section 3 describes the core of the work of this paper, detailing the methodological engineering approach and the main ontology constituents, while providing usage examples and evaluation details. Section 4 informs the reader on the documentation and publishing activities for NAct, while Section 5 provides a conclusion and describes already ongoing future work.

2. Related work

Previous research yielded several key European and International food and nutrient databases complete with few pre-existing nutritional ontologies. The list of food and activity databases is non-exhaustive and thus will not be listed. We will however mention the McCance and Widdowson food database [6] and the Compendium of Physical Activities [7], which were deemed by the domain experts as the vastest and most adequate databases adhering to European nutritional, health and well-being standards. These databases subsequently inspired, to an extent (re top level foods and activities), the respective NAct aspects.

However, the purpose of NAct is not to exhaustively model all possible foods/ ingredients and activities in a mere list or even taxonomy, but rather provide a serve as a robust and intelligent backbone for a knowledge-based AI recommendation system. Such a system would only benefit from a well structured, well defined ontology to serve as the TBox[2] for subsequent logical inference of suitable nutritional and activity directives to users of a smart healthy living platform. The problem and its requirements are detailed in Section 3.1.

Of the few relevant ontologies that were identified within the literature, most lacked rich semantic correlations, or do not model key components that are required for the purposes of the knowledge-based expert AI system that employs NAct, since they

---

[2] Terminological Box

serve a different purpose than the scope of NAct. For example, seminal works such as FoodOn [8], ONS [9], FOBI [10] and CDNO [11] are of relatively shallow expressivity, focusing on extensively modelling, structuring and relating food products and their biochemical role for data retrieval, while lacking enough axiomatic interconnections that allow for advanced recommendation of healthy dietary directives. Comparably, several ontologies that deal with biochemical properties of foods such as ONE [12] and FIDEO [13] bear similar expressivity and are focused on the biochemical properties of foods in relation to very particular health issues (respectively, epidemics and drug reactions) that eschew from the general healthy dietary directives domain.

The two most relevant ontologies to the proposed problem, are the Food Ontology (FOKB) [14] and the HeLiS ontology [15]. Both model food types and nutritional information about them, with the FOKB delving into details about properties of food products, including additives and governing agents (e.g. *anticaking*, *antifoaming*), while HeLiS modelling foods and nutrients as well as physical activities.

The main purpose of FOKB is to serve as the background knowledge to determine side effects of compound and manufactured foods to users allergies and some medical conditions, which is relevant to one of NAct's main requirements, i.e. consider allergies and medical conditions in healthy living recommendations. In the context of NAct, this pertains to a core food/ ingredient layer, with relevant connections to medical conditions. FOKB on the other hand delves into the specifics of properties after a produce has been processed (e.g. *additives* etc.). Most importantly, FOKB lacks semantics about particular nutrient-to-food relations that may be used to promote nutritional best practices, as well as any connection to physical activities in relation to conditions. Lastly, FOKB does not interrelate produce information with any nutritional and well-being user goals.

As far as HeLiS is concerned, this is the only other ontology known to date that includes both nutritional and physical activities information. It also includes classifications of nutrients, which FOKB lacks. However, there is a distinct lack of axiomatic interconnection between food types and nutrients or physical activities and properties that affect undertaking these activities (e.g., a medical condition). Rather, those facets are merely presented as a hierarchy of concepts under which a plethora of predetermined individuals are instantiated (e.g. particular, non-updateable activities and undefined nutrient specifics, e.g. *Alcohol_000* under alcohol). There is no freedom to instantiate anything else under these classes, whereas the expert system that employs NAct aims to be able to instantiate any foods, activities and any other information under its core set of abstract entities. Most prominently, HeLiS lacks relations or axioms at the schema basis to liaise the aforementioned information (nutrients, foods, activities) with each other either with respect to particular medical conditions, allergies or with dietary/well-being goals.

However, both of these ontologies have inspired technical aspects of the engineering of NAct, as per relevant shared objectives, i.e., the foods, nutrients and activities structure.

3. Methodology

Engineering the NAct ontology followed the Methontology [16] methodology. This pertains to seven stages: specification, knowledge acquisition, conceptualisation, integration, implementation, evaluation and documentation. Each stage's developments per the NAct ontology are detailed in the following subsections.

Methontology was elected due to empirical affirmation in past ontology engineering endeavors that the method facilitates the process of creating a new ontology in a collaborative manner by multi-disciplined domain experts and ontology engineers. It is also found to enable pragmatic observations and requirements gathering and, consequently, an efficient process to maintain and evolve the ontology.

3.1. Specification

This phase documents the purpose of the ontology, its semantic expressivity and its scope. The objectives behind engineering NAct can be summarised in the following:

- Model in a slim and holistic manner food-specific nutritional information and activity-specific well-being information.
- Model nutritional and well-being user goals and relate them with nutritional and well-being information.
- Model medical conditions, allergies, intolerances, deficiencies and lifestyle dietary choices and related them with nutritional and well-being information.
- Model properties that define specificities of the aforementioned relationships that aid in the selection of appropriate meals and physical activities for a given person.

The core engineering scope behind this objective is to refrain from a non-exhaustive listing of all foods, activities and their respective detailed information as can be found in existing databases, but rather abstract and generalise as much as possible to basic food and activity types and the most prominent of their respective nutritional and well-being impact, in order to ensure tractability and at the same time decidability in the inference process. To this end, expert-provided information has been distilled into a set of well-defined concepts, relations between them and complex rules that connect them.

The expressivity chosen, to align also with the reasoning capacities of the reasoning component that employs NAct, namely the LiFR fuzzy reasoner [17], lies within the OWL 2 RL[3] fragment.

3.2. Knowledge Acquisition

The foundations of the NAct ontology is a wealth of evidence-based information gathered from nutrition scientists, medical experts and scientists with a vast expertise in kinesiology and rehabilitation sciences within the PROTEIN EU[4] project consortium. The concepts of the ontology were based on the information gathered from the health professionals, which was then connected with the expertise of consortium engineers in semantics, AI/expert systems and logic-based inferencing.

Relations and rules have been developed within the project, to enable end users to achieve their nutritional goals and to relate the nutrients and medical conditions of consumers with the nutritional requirements within the different PROTEIN user groups, such as the overall healthy population, as well as patients with obesity, cardiovascular disease (CVD), Type 2 Diabetes and iron deficiency.

---

[3] `https://www.w3.org/TR/owl2-profiles/`

[4] `https://protein-h2020.eu/`

As aforementioned, following a review of the current literature no databases or ontologies currently exist that adequately model the correlations between the mode of physical activity (PA) and nutrition with specific dietary and well-being goals or with medical conditions. Furthermore, no ontologies that specify PA or nutritional rules for particular conditions and diets such as the ones pertaining to the PROTEIN project, as discussed previously were identified. Therefore, the problem at hand required the novel conception of a condition and goals-specific ontology in relation to nutritional and PA aspects.

Overall, NAct has been developed through close and immediate collaboration between the ontology engineering experts and the medical/ nutrition/ PA experts within the PROTEIN consortium, following the analysis of various European databases standards and guidelines (mentioned in Section 5). Several case-based workshops were held during the winter and spring of 2020, discussing ontology requisites and trade-offs, in terms of foods, physical activities, medical conditions and user goals and designing the rules that would interrelate these facets. Workshop results were recorded on an online spreadsheet tracker and used to put experts' knowledge in a machine-understandable formalization under the NAct ontology.

After an initial set of five workshops per user group (overall population, obesity/overweight population, athletes, iron deficiency, type 2 diabetes) an iterative process of engineering the ontology and presenting it to experts for revision was followed, which resulted in the first version of NAct.

3.3. *Conceptualisation: NAct in Depth*

This phase deals with the glossary of terms that comprise of the core ontology vocabulary, identifying all the useful domain knowledge and its semantics, as well as the inference rules that will guide a personalized food and activity recommendation system. This vocabulary and rules were a product from the crystallization of all the information that was gathered from the databases and other ontologies and relevant vocabularies examined, but also from the important intangible knowledge offered by the experts in the dedicated virtual and physical workshops held between the experts and the ontology engineers.

To this end, Figure 1 represents the top level concepts of the NAct ontology.

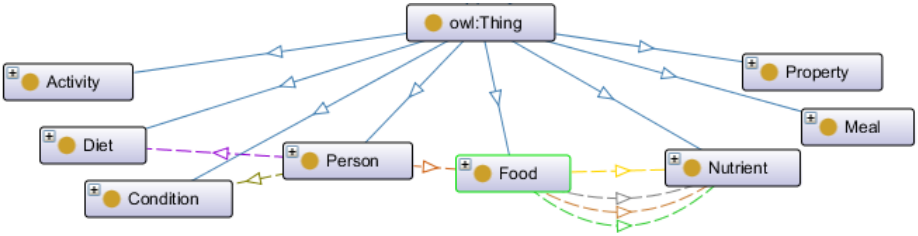


Figure 1. The top level concepts of the NAct ontology.

Activity (Fig. 2) models a hierarchy of physical activities. This hierarchy was inspired by the Compendium of Physical Activities [7], as well as by the activities of the HeLiS ontology, while it was revised by domain experts as per its compliance to European well-being directives and minimised to the optimal granularity through collaboration of ontology engineers with domain experts.

Condition (Fig. 2) covers the main medical conditions pertaining to the specific patient user groups of PROTEIN, i.e. cardiovascular disease, diabetes and obesity. In addition, some other prominent conditions were included for the general population. Most importantly, a complete set of allergies, intolerances and deficiencies were modelled, to cover the most important dietary and exercise restrictions and needs for all users employing a healthy lifestyle directives recommendation system.

Diet (Fig. 2) includes a set of dietary restrictions that may affect the food choices of users. It was decided by the domain experts that the ontology's focus should not align with preferential (e.g. Mediterranean) or commercial/popular (e.g. Atkins) diets, but rather maintain a high individualisation level per each user and their respective needs, i.e. combining preferences and needs in a flexible way rather than relying on diet templates. For this reason, only particular lifestyle or condition-related choices (e.g. vegan, halal) that provide specific dietary restrictions were modeled.

Meal and ***Person*** (Fig. 2) consist of basic classifications of meal types (e.g. breakfast) and of users (e.g. overweight adult). The former serves as a filter for the final decisions of the reasoning-based nutrition and activity AI advisor. The latter correlates to specific nutritional and well-being guidelines, as defined in particular axioms (described further on).

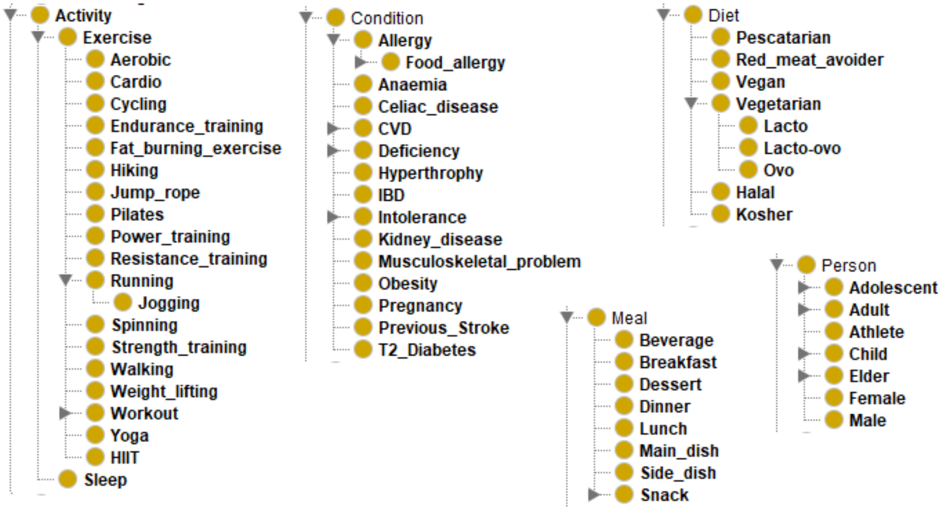


Figure 2. Activity, Condition, Diet, Meal and Person.

Food (Fig. 3) comprises a non-exhaustive hierarchy of principle foods. This is the main point where an important trade-off needed to be made in comparison to the plurality of detailed variations of foods that exist in existing food databases: the granularity must not be too deep, rather the most universally commonly ingredients of meals need to be included in their basic form, and for all those primary components not included, comprehensive food categories need to be available, so that undefined meal ingredients can be classified under the categories. Only a minimal set of *compound foods*[5], common

---

[5] By compound foods, we denote foods that pertain a composition of basic ingredients.

in European diets (e.g. pasta, bread) or popular in particular lifestyle diets (e.g. seitan, falafel for vegetarians) were modeled. This vocabulary was primarily inspired by the McCance and Widdowson database [6], while engineers have taken into account the related HeLiS and FOKB classes, while the final sub-hierarchy was supervised and adapted by the experts based on the European Commission's *Food-Based Dietary Guidelines* [18].

Nutrient (Fig. 3) is a crucial sub-hierarchy in NAct. This sub-hierarchy was constructed based on the directives of the European nutritional guidelines [19]. It serves as the means to correlate specific foods and food groups with nutrients and subsequently determine the most and least nutritionally valuable meals per each individual user as per their specific (explicit) preferences and conditions. What drives the personalization system's decisions under NAct's scope is the finite set of nutrients, not an exhaustive list of foods and a voluminous set of instances denoting properties of each individual food, thus boosting both the system's flexibility as well as the recommender's computational efficiency.

A well-structured and meticulous hierarchical structure for both foods and nutrients was imperative in the scope of achieving NAct's purposes. The correlation among foods, food super-groups, nutrients and nutrient super-groups, conditions and goals is the core for determining the suitability of meals for each specific personalized nutrition application user.

Lastly, ***Property*** (Fig. 3) contains several types of important properties that need to be correlated with relevant nutritional and activity suggestions, like activity properties (e.g. level and intensity of activities, food and nutrient properties, food attributes, cooking/ preparation styles, etc., but most prominently ***Goals*** and ways to ensure their achievement. Goals influence the core of the nutrition and activity AI advisor and were provided by the domain experts based on multi-disciplinary empirical evidence and observations.

3.3.1. Relations and Rules

Pivotal to the aforementioned correlations between concepts was the definition of a minimal and meaningful set of binary relations (i.e. object properties). These relations were used in rules that drive the reasoning-based advisor's inference process. Rules in NAct comprise GCIs[6] and non-GCIs axioms.

The defined relations and an example of inference rules are displayed in Figures 4 and 5. Most relations are assigned with a domain and/or range that define the semantic relation they support. For instance, the property *"highIn"* has *Food* as domain and *Nutrient* as range. This means that *Foods* (and only foods) may be *highIn* one or more *Nutrients* (and only nutrients).

One of the most important set of object properties is the food-to-nutrient relations (highIn, lowIn, containsNutrient) and subsequent rules. In order to gather knowledge regarding these rules, the nutritional correlation of all ontology foods was examined following the European Commision's Food Claims[7]. Consequenly, relevant inference rules were automatically extracted based on concentration of nutrients in relevant foods and food types.

---

[6] General Concept Inclusion

[7] `https://ec.europa.eu/food/safety/labelling_nutrition/claims/nutrition_claims_en`

Figure 3. Food, Nutrient and Property.

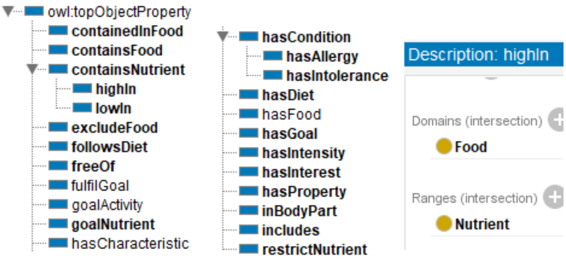
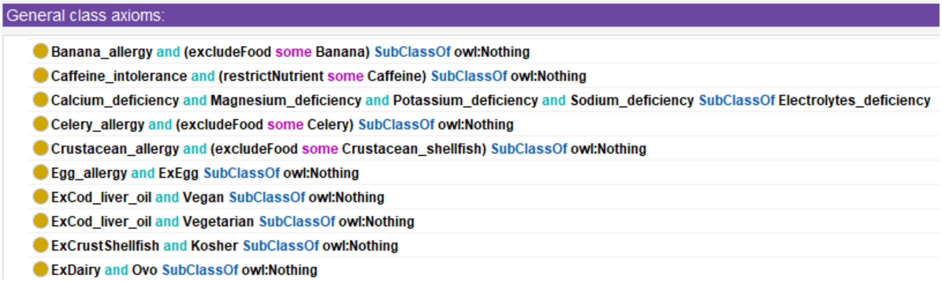


Figure 4. NAct relations.


General class axioms:

Banana_allergy and (excludeFood some Banana) SubClassOf owl:Nothing

Caffeine_intolerance and (restrictNutrient some Caffeine) SubClassOf owl:Nothing

Calcium_deficiency and Magnesium_deficiency and Potassium_deficiency and Sodium_deficiency SubClassOf Electrolytes_deficiency

Celery_allergy and (excludeFood some Celery) SubClassOf owl:Nothing

Crustacean_allergy and (excludeFood some Crustacean_shellfish) SubClassOf owl:Nothing

Egg_allergy and ExEgg SubClassOf owl:Nothing

ExCod_liver_oil and Vegan SubClassOf owl:Nothing

ExCod_liver_oil and Vegetarian SubClassOf owl:Nothing

ExCrustShellfish and Kosher SubClassOf owl:Nothing

ExDairy and Ovo SubClassOf owl:Nothing

Figure 5. NAct GCIs.

Several rules have been created based on obvious correlations (e.g. foods that cause specific allergies) or more implicit correlations defined by the experts, for each condition in relation to foods, nutrients, activities and their relevant properties, in order to ensure that the user will not be recommended with foods/activities that they need to avoid based on their conditions or that they need to consume more of and activities that they need to undergo less or more of. An example would be $Gluten \sqcap \exists restrictNutrient.Gluten\_Intolerance \sqsubseteq \bot$, which restricts from a Gluten intolerant user's diet any foods that contain Gluten.

Another example is the set of axioms pertaining to Goals. Goals may be explicitly declared through the user profile or be implicitly derived from the inference engine, based on relevant rules that have been modelled within the ontology, e.g. $Iron\_Deficiency \sqsubseteq \forall hasGoal.IncreaseIronIntake$. This rule defines that every person that has iron deficiency always has an implicit goal to increase their iron intake.

Based on the above, the inference engine will promote Foods that are defined to be high in the nutrient Iron, since another rule included in the ontology is that $IncreaseIronIntake \sqsubseteq \forall highIn.Iron$.

Lastly, goals contain even more complex rules, such as $Adult \sqcap Athlete \sqcap Muscle\_gain \sqsubseteq \forall highIn.Protein \sqcap \forall highIn.Carbohydrates$, which denote that if the user is an Adult and an Athlete, and their (explicitly declared in the user profile) goal is Muscle Gain, then they should increase Protein intake and Carbohydrates intake (therefore consume more foods that are rich in these nutrients).

3.4. Integration: NAct in Action

NAct has been integrated with the knowledge-based expert system of the PROTEIN project, namely the AI Advisor, which employs the LiFR fuzzy reasoner for inferring the optimal meals, restaurant menu items and physical activities to recommend to a given user, based on this user's dietary and medical profile.

The recommendation system matches the explicit user-declared profiles (per user) against all possible meal and activity options available in the PROTEIN system, taking into account the nutritional, biomedical and physical activity background knowledge modeled in NAct.

In order to achieve this, semantic profiles of recommendation candidates (meals/ restaurant menu, activities) and user profiles are automatically created from the list of ingredients of available meals and activities, as well as from the list of each user's dietary and medical premises. These profiles add candidate- and user-pertinent axioms to the TBox (most of the TBox comprising of NAct), while providing the ABox to complete the matching problem's KB.

These semantic profiles' purpose is dual: (a) transform the meals/activities and user profiles into reasoner-understandable formalizations, but most importantly (b) impose implied concept and relation instances, beyond the ones explicitly available in the profiles, in order to instigate a query process in the inference mechanism based on the ontology's model. An example of a semantic candidate (meal or activity) profile is shown in Table 1. Similarly, a semantic user profile example can be seen in Table 2.[8]

---

[8] It should be noted that LiFR supports fuzzy concept assertions, therefore it can accept concept instance degrees such that $\langle a : C \bowtie d, d \in [-1.0, 1.0]$ and preference weights $w \cdot C, w \in [0.0, 1.0]$. In crisp cases, $d \geq 1.0$ and $w = 1.0$ is implied. Such clauses will be used in the ABoxes and the inference examples that will follow further on.

Table 1. Semantic candidate profile example.

$\exists includes.(Spinach \sqcap ...) \sqsubseteq Breakfast\_C2\_1500$	Constituents of the candidate. In this case, a conjunction of this meal's ingredients
$\langle sp\_1 : Spinach \rangle$	Instance sp_1 of type Spinach
$\langle candidate, sp\_1 : includes \rangle$	The candidate meal contains sp_1 of type Spinach (for preference check)
$\langle sp\_1, nutr : containsNutrient \rangle$ [7]	Look for the nutrients that all classes which sp_1 asserts contain
$\langle sp\_1, nutr : highIn \rangle$ [7]	Look for what nutrients all classes that sp_1 asserts are high in
$\langle sp\_1, nutr : lowIn \rangle$ [7]	Look for what nutrients all classes that sp_1 asserts are low in
$\langle ingr, sp\_1 : containsFood \rangle$	Look for compound foods that contain, as ingredient(s), all classes that sp_1 asserts
$\langle user, sp\_1 : excludeFood \rangle$	Look for asserted premises for which all classes that sp_1 asserts must be excluded

Table 2. Semantic user profile example

$\exists hasInterest.(Vegetable \sqcap ...) \sqsubseteq uid75$	What the user likes to eat or do (activity-wise)
$\exists hasInterest.(Yoghurt \sqcap ...) \sqsubseteq uid75\_dis$	What the user doesn't like to eat or do (activity-wise)
$uid75 \sqcap uid75\_dis \sqsubseteq \bot$	Disjoint user likes and dislikes
$0.89 \cdot Vegetable$	Preference weight
$\exists fulfilGoal.ImplicitGoal \sqsubseteq uid75$	Default user profile axiom: search for implicit goals inflicted by user premises
$\langle user : uid75 \rangle$	User instance
$\langle user : Iron\_Deficiency \rangle$	User has iron deficiency
$\langle user : Banana\_Allergy \rangle$	User has banana allergy
$\langle user, goal : hasGoal \rangle$[8]	Look for goals that can should fulfilled for this specific user
$\langle nutr, goal : goalNutrient \rangle$ [8]	Look for which goals an asserted nutrient can fulfil
$\langle act, goal : goalActivity \rangle$ [8]	Look for which goals an asserted activity can fulfil
$\langle candidate, goal : fulfilGoal \rangle$ [8]	Look if the candidate (meal, activity) fulfils a goal
$\langle user, nutr : restrictNutrient \rangle$	Look if there is any premise inferred that causes the restriction of a nutrient for this user

3.4.1. Usage Examples

This section details the main test scenario of the inference process, validating the capacity of NAct to yield appropriate recommendations and restrictions based on of user-related information.

---

[7] It is anticipated that for the next expansion of LiFR, fuzzy relation assertions and weighted relations will be included and *highIn, lowIn, containsNutrient* will be assigned with different weights and thus assertions for them will result to different entailment degrees in the inferred model. Until then, only the *highIn, containsNutrient* instances are actually included in the employed ABox, as they denote a significant impact of a nutrient in particular user goals or deficiencies.

[8] In the premises of the recommendation problem at hand, it was decided that any goal holding true for the candidate is sufficient to produce a match, therefore only one *goal* instance is employed. If one wants to use NAct to discern between fulfilled goals, we encourage using an enumeration of *goal* instances, such as $goal\_1, goal\_2$, etc. for each explicit goal or medical condition that is included in the user profile. To this end, the user profile must include a set of the referenced relation instances, one per $goal\_X$ instance.

Table 3. Example of candidate fulfilling user preference

Ontology axioms		**User premises**		**Candidate facts**	
$Spinach \sqsubseteq Vegetable$	(1a)	$\exists hasInterest.Vegetable \sqsubseteq uid75$	(2a)	$\exists includes.Spinach \sqsubseteq Breakfast\_C2\_1500$	(3a)
$includes^{-} \equiv hasInterest$	(1b)	$0.89 \cdot Vegetable$	(2b)	$\langle sp\_1 : Spinach \rangle$	(3b)
		$\langle candidate, sp\_1 : includes \rangle$	(2c)		

Inference			
(1a) $:\Leftrightarrow$	$vegetable(x) \leftarrow spinach(x)$		
$\because$ (3b)	$vegetable(x) \leftarrow spinach(sp\_1)$		
$\models$	$vegetable(spi\_1)$ $\Leftrightarrow vegetable(spi\_1) \geq 1.0$	(4)	The meal contains a vegetable
(2b) $:\Leftrightarrow$	$vegetable(x) \geq 1.0 \cdot 0.89$		
$\because$ (4)	$vegetable(spi\_1) \geq 1.0 \cdot 0.89$		
$\models$	$vegetable(spi\_1) \geq 0.89$	(5)	The fact that the inferred model contains a vegetable is important to the user by 0.89
(1b) $:\Leftrightarrow$	$\begin{cases} includes(x,y) \leftarrow hasinterest(x,y) & (1bi) \\ hasinterest(x,y) \leftarrow includes(x,y) & (1bii) \end{cases}$		
[1bii] $\because$ (2c)	$hasinterest(x,y) \leftarrow includes(candidate, spi\_1)$		
$\models$	$hasinterest(candidate, spi\_1)$	(6)	The user may be interested in a candidate that includes *spi*_1, i.e. a spinach instance
(2a) $:\Leftrightarrow$	$uid75(x) \leftarrow hasinterest(x,y), vegetable(y)$		
$\because$ (5), (6)	$uid75(x) \leftarrow hasinterest(candidate, spi\_1), vegetable(spi\_1) \geq 0.89$		
$\models$	$uid75(candidate) \geq 0.89$		The given candidate meal satisfies the user profile, with a suitability degree of 0.89

One aspect pertains to meals and/or activities that *should* be recommended to a given user, because they may satisfy the user's preferences (Table 3) or because they may satisfy a particular user goal (Table 4). In the subsequent examples, the DL (Description Logics) axioms and instances will be translated to propositional logic clauses demonstrating the inference process.

The other major aspect in NAct's usage pertains to *rejections* of foods and/or activities. Rejections are of the most important operations of the recommendation system. They determine whether a candidate must absolutely not be recommended or even presented to the user. They are evoked whenever a logical contradiction (refutation) occurs when reasoning over a candidate. This happens in two cases:

A. When an ingredient in a meal or a type of activity has been explicitly declared by the user as one of their disinterests.

B. When an ingredient in a meal or a type of activity comes with in contrast with one of the user's characteristics (e.g. meat in case of a vegetarian user).

C. When a nutrient of an ingredient in a meal or a property of an activity (e.g. high intensity running) is actively prohibited given the user's medical condition(s).

Table 4. Example of candidate fulfilling goal

Ontology axioms		**User premises**		**Candidate facts**	
$Spinach \sqsubseteq \forall highIn.Iron$	(1a)	$\exists fulfilGoal.ImplicitGoal \sqsubseteq uid7$	(2a)	$\exists includes.Spinach \sqsubseteq Breakfast\_C2\_1500$	(3a)
$Iron\_Deficiency \sqsubseteq \forall hasGoal.IncreaseIronIntake$	(1b)	$\langle user : Iron\_Deficiency \rangle$	(2b)	$\langle sp\_1 : Spinach \rangle$	(3b)
$IncreaseIronIntake \sqcap \exists goalNutrient.Iron \sqsubseteq ImplicitGoal$	(1c)	$\langle user, goal : hasGoal \rangle$	(2c)	$\langle sp\_1, nutr : highIn \rangle$	(3c)
		$\langle nutr, goal : goalNutrient \rangle$	(2d)		
		$\langle candidate, goal : fulfilGoal \rangle$	(2e)		

Inference			
(1a): $\Leftrightarrow$	$iron(y) \leftarrow highin(x,y), spinach(x)$		
$\because$ (3b)	$iron(y) \leftarrow highin(spi\_1, nutr), spinach(spi\_1)$		
$\models$	$iron(nutr)$	(4)	The meal contains the nutrient Iron
(1b): $\Leftrightarrow$	$increaseironintake(y) \leftarrow hasgoal(x,y), iron\_deficiency(x)$		
$\because$ (2b), (2c)	$increaseironintake(y) \leftarrow hasgoal(user, goal), iron\_deficiency(user)$		
$\models$	$increaseironintake(goal)$	(5)	The goal *increase iron intake* is inferred as true for this KB
(1c): $\Leftrightarrow$	$\begin{cases} IncreaseIronIntake \sqcap A \sqsubseteq ImplicitGoal & (1ci) \\ \exists goalNutrient.Iron \sqsubseteq A & (1cii) \end{cases}$		
(1cii) : $\Leftrightarrow$	$a(y) \leftarrow goalNutrient(x,y), Iron(x)$		
$\because$ (2d), (4)	$a(y) \leftarrow goalNutrient(nutr, goal), iron(nutr)$		
$\models$	$a(goal)$	(6)	The nutrient needed to fulfill this goal holds true for this KB
(1ci) : $\Leftrightarrow$	$implicitgoal(x) \leftarrow increaseironintake(x), a(x)$		
$\because$ (5), (6)	$implicitgoal(x) \leftarrow increaseironintake(goal), a(goal)$		
$\models$	$implicitgoal(goal)$	(7)	An implicit goal is satisfied for this user
(2a) : $\Leftrightarrow$	$uid75(x) \leftarrow fulfilgoal(x,y), implicitgoal(y)$		
$\because$ (2e), (7)	$uid75(x) \leftarrow fulfilgoal(candidate, goal), implicitgoal(goal)$		
$\models$	$uid75(candidate)$		The given candidate meal satisfies the user profile; the suitability degree is implied to be 1.0

Due to length restrictions, a complete rejection example will not be detailed. Axioms in the ontology that imply $\perp$ (owl:Nothing), e.g. $Banana\_allergy \sqcap \exists excludeFood.Banana \sqsubseteq \perp$ and $Gluten\_Intolerance \sqcap \exists restrictNutrient.Gluten \sqsubseteq \perp$ are designed exactly to cause such refutations whenever relevant foods, nutrients, activities or other properties that come in contrast with the user profile are inferred.

In the same mentality, in terms of user preferences, the disjointness axiom $uidX \sqcap uidX\_dis \sqsubseteq \perp$ of Table 2 is employed to cause such refutations. Therefore, whenever a meal or activity fulfills a user interest (with one or more ingredients for the meal case), while at the same time another candidate (e.g. ingredient) fulfills the disinterests, the reasoner will issue a refutation, causing for the said meal to be rejected for this user from the list of candidates.

3.5. Implementation

NAct is an OWL ontology, falling in the OWL 2 RL expressivity fragment, as mentioned before. Thus it leverages rich expressivity and computational efficiency in order to enable robust logic-based inferencing for content recommendation, but at the same time reduce the computational cost. Throughout its lifecycle, it has been engineered using the Protégé [11] ontology editor.

3.6. Evaluation

Based on a pre-defined pool of >1400 expert-defined meals and >50 physical activities of different intensity levels available in the PROTEIN platform, experiments were held using the LiFR reasoner with over 70 synthetic user profiles including one or more allergies, deficiencies, intolerances, diet choices and medical conditions, with several combinations thereof, in a pre-release phase of the first version of the PROTEIN system. Meals, activities and user profiles were semantically transcoded as described in Section 3.4.

Through these experiments, NAct has been validated technically in terms of logical Consistency and Completeness, Soundness and Decidability as well as of Computational Efficiency [20], [21]. The tests were held both by technical staff as well as the domain experts, simulating their patients and clients.

NAct has been found to be decidable (sound & complete) - complete in the sense that any expression that is logically implied by the KB[12] that includes NAct and the meal/activity and user profiles as previously described, can be derived. It is also consistent - in the sense that only purposeful contradictions arise during the reasoning process.

In terms of computational efficiency, the results depend on the respective high computational efficiency of the LiFR reasoner, as described in [17] and vary according to the computational capacities of the machine that runs the inference service. In any case, memory consumption is insignificant (re LiFR), while matching a single meal's semantic profile with a given user's semantic profile on top of NAct takes 1-3 seconds on a Intel Core i5 on 3.3GHz, depending on the number of instances in the meal and user profiles.

NAct however pends validation in the ongoing PROTEIN pilots in terms of Consistency, Completeness[13] and Conciseness, to what it concerns fully covering the well-being recommendation needs of the users of the project.

4. Availability and Documentation

NAct is publicly available under the Creative Commons Attribution-ShareAlike License (version 3.0)[14], under a persistent PURL URI, namely `http://purl.org/nact`. The

---

[11] `https://protege.stanford.edu/`

[12] As per the definition of Logical Completeness of [21]

[13] In the sense of recall

[14] `http://creativecommons.org/licenses/by-sa/3.0/`

ontology is published on GitHub, in a dedicated project and repository[15], while the ontology specification and documentation (LODE [22] version) web page[16] will be permanently maintained through GitHub pages.

Two PROTEIN project deliverables serve as the means to document the first version and subsequent evolutions of the PROTEIN ontology. Moreover, technical documentation was provided by means of the OWLDoc[17] ontology documentation producing tool.

Furthermore, in NAct's website a public summation of the developments of each release is maintained, accompanied by formal documentation of the ontology's contents produced via the LODE [22] tool. The OWLDoc documentation is also available on the site.

5. Conclusions and Future Work

This paper presented the first version of the novel NAct ontology, which innovatively combines evidence-based and consolidated EU standards-based nutritional, medical and preferential elements for advanced individualization of meal and physical activity recommendations in an intelligent AI-based healthy lifestyle system.

As the work presented comprises the first version of such an expert-based system, only having undergone synthetic trials and expert evaluation, and is yet to be tested in real-world pilots, evolution of the ontology is expected in the near future.

It is for example a known fact to the engineers and experts that not all prominent inference rules that can be modeled for the domain in question are included in this first version of the ontology, however the first pilots aim to reveal redundancies of the first version and pinpoint the most important rules that have not yet been included in the system. One major action point for experts and ontology engineers, taking place before summer 2021, will be to add several more relevant rules to the ontology relating more medical conditions to physical activities and their properties (e.g. intensity).

In addition, later extensions will also delve in formally defining the semantics of the modelled entities, as well as in providing mappings to entities of similar semantics in seminal related ontologies and/or vocabularies.

The ontology engineers and domain experts will continue their collaboration to extend and revise the novel ontology - at least - throughout the PROTEIN project's lifecycle, following own observations while using the system, but most importantly based on end users' evaluation in the first pilots.

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme, grant *n^o^* 817732, PROTEIN (”PeRsOnalized nutriTion for hEalthy livINg”).

---

[15] `https://github.com/nutritionactivityontology/nact`

[16] `https://nutritionactivityontology.github.io/nact/`

[17] `https://protegewiki.stanford.edu/wiki/OWLDoc`

References

[1] (EFSA) EFSA. Tools for critically appraising different study designs, systematic review and literature searches. EFSA supporting publications. 2015;12(7):836E.

[2] Hariton E, Locascio JJ. Randomised controlled trials—the gold standard for effectiveness research. BJOG: an international journal of obstetrics and gynaecology. 2018;125(13):1716.

[3] Schwingshackl L, Chaimani A, Hoffmann G, Schwedhelm C, Boeing H. A network meta-analysis on the comparative efficacy of different dietary approaches on glycaemic control in patients with type 2 diabetes mellitus. European journal of epidemiology. 2018;33(2):157-70.

[4] Theodoridis T, Solachidis V, Dimitropoulos K, Gymnopoulos L, Daras P. A survey on AI nutrition recommender systems. In: Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments; 2019. p. 540-6.

[5] Wilson-Barnes S, Gymnopoulos L, Dimitropoulos K, Solachidis V, Rouskas K, Russell D, et al. PeRsOnalised nutriTion for hEalthy livINg: The PROTEIN project. Nutrition Bulletin. 2021;46(1):77-87.

[6] McCance RA, Widdowson EM. McCance and Widdowson's the Composition of Foods. Royal Society of Chemistry; 2014.

[7] Shephard R. compendium of physical activities: a second update of codes and MET values. Yearbook Of Sports Medicine. 2011;2012:126-7.

[8] Dooley DM, Griffiths EJ, Gosal GS, Buttigieg PL, Hoehndorf R, Lange MC, et al. FoodOn: a harmonized food ontology to increase global food traceability, quality control and data integration. npj Science of Food. 2018;2(1):1-10.

[9] Vitali F, Lombardo R, Rivero D, Mattivi F, Franceschi P, Bordoni A, et al. ONS: an ontology for a standardized description of interventions and observational studies in nutrition. Genes & nutrition. 2018;13(1):1-9.

[10] Castellano-Escuder P, González-Domínguez R, Wishart DS, Andrés-Lacueva C, Sánchez-Pla A. FOBI: An ontology to represent food intake data and associate it with metabolomic data. Database. 2020;2020.

[11] Andrés-Hernández L, Baten A, Azman Halimi R, Walls R, King GJ. Knowledge representation and data sharing to unlock crop variation for nutritional food security. Crop Science. 2020;60(2):516-29.

[12] Yang C, Ambayo H, De Baets B, Kolsteren P, Thanintorn N, Hawwash D, et al. An ontology to standardize research output of nutritional epidemiology: from paper-based standards to linked content. Nutrients. 2019;11(6):1300.

[13] Bordea G, Nikiema J, Griffier R, Hamon T, Mougin F. FIDEO: Food Interactions with Drugs Evidence Ontology. In: 11th International Conference on Biomedical Ontologies; 2020. .

[14] Çelik D, Elçi A, Akcicek R, Gökçe B, Hürcan P. A safety food consumption mobile system through semantic web technology. In: 2014 IEEE 38th International Computer Software and Applications Conference Workshops. IEEE; 2014. p. 348-53.

[15] Dragoni M, Bailoni T, Maimone R, Eccher C. Helis: An ontology for supporting healthy lifestyles. In: International Semantic Web Conference. Springer; 2018. p. 53-69.

[16] Fernández-López M, Gómez-Pérez A, Juristo N. Methontology: from ontological art towards ontological engineering. 1997.

[17] Tsatsou D, Dasiopoulou S, Kompatsiaris I, Mezaris V. LiFR: A lightweight fuzzy DL reasoner. In: European semantic web conference. Springer; 2014. p. 263-7.

[18] EFSA Panel on Dietetic Products, Nutrition, and Allergies (NDA). Scientific opinion on establishing food-based dietary guidelines. EFSA Journal. 2010;8(3):1460.

[19] (EFSA) EFSA. Dietary reference values for nutrients summary report. Wiley Online Library; 2017.

[20] Gómez-Pérez A. Evaluation of ontologies. International Journal of intelligent systems. 2001;16(3):391-409.

[21] Obrst L, Ceusters W, Mani I, Ray S, Smith B. The evaluation of ontologies. In: Semantic web. Springer; 2007. p. 139-58.

[22] Peroni S, Shotton DM, Vitali F. Making Ontology Documentation with LODE. In: I-SEMANTICS (Posters & Demos). Citeseer; 2012. p. 63-7.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210378

Capturing the Basics of the GDPR in a Well-Founded Legal Domain Modular Ontology

Mirna EL GHOSH [a,1] and Habib ABDULRAB [a]
[a] *Normandie Université, INSA Rouen, LITIS, 76000 Rouen, France*

Abstract. The primary goal of the General Data Protection Regulation (GDPR) is to regulate the rights and duties of citizens and organizations over personal data protection. Implementing the GDPR is recently gaining much importance for legal reasoning and compliance checking purposes. In this work, we aim to capture the basics of GDPR in a well-founded legal domain modular ontology named OPPD (Ontology for the Protection of Personal Data). Ontology-Driven Conceptual Modeling (ODCM), ontology layering, modularization, and reuse processes are applied. These processes aim to support the ontology engineer in overcoming the complexity of the legal knowledge and developing an ontology model faithful to reality. ODCM is used for grounding OPPD in the Unified Foundational Ontology (UFO). Ontology modularization and layering aim to simplify the ontology building process. Ontology reuse focuses on selecting and reusing Conceptual Ontology Patterns (COPs) from UFO and the legal core ontology UFO-L. OPPD intends to overcome the lack of a representation of legal procedures that most ontologies encountered. The potential use of OPPD is proposed to formalize the GDPR rules by combining ontological reasoning and Logic Programming.

Keywords. GDPR, well-founded ontologies, Ontology-Driven Conceptual Modeling, ontology modularization, ontology reuse, Conceptual Ontology Patterns, Logic Programming

1. Introduction

The General Data Protection Regulation (GDPR)[2] is a European Union Regulation established in 2018 [1]. The GDPR regulates the rights and duties of citizens and organizations regarding the protection of personal data. It contains obligations concerning storing, processing, collecting, and disclosing data [2]. The implementation of GDPR is recently gaining much importance aiming to apply the Regulation in organizations [3,4,5]. Organizations seek to comply with the Regulation using technical measures to ensure that personal data processing follows GDPR [2]. In this context, a variety of approaches have been recently proposed such as AI-based [6,7,8], model-based [9], semantic annotation of text [10], and ontologies [11,12,13]. In this work, we are interested in ontologies. In

---

[1] Corresponding Author: El Ghosh Mirna, Normandie Université, INSA Rouen, LITIS, 76000 Rouen, France; E-mail: mirna.elghosh@insa-rouen.fr

[2] https://gdpr-info.eu/

the legal domain, ontologies are considered to establish the missing link between legal theory and AI & law [14]. They are defined as *generalized conceptual models of specific parts of the legal domain* [15]. They provide *stable foundations for knowledge representation* in this domain [15]. Legal ontologies have been developed and used for legal knowledge management and as knowledge bases in legal knowledge systems [16]. However, modeling legal knowledge is challenging due to the legal domain's complexity represented by regulations and legal rules. In the legal domain, legal conceptual knowledge is closely related to the language used in legal documents, which is usually considered complicated by non-experts [17]. Legal rules and standards are written, for the most part, in ordinary language containing ambiguities [18]. Specifically, we cite the incomplete definition of the law's legal concepts (e.g., consent, right, duty, etc.) [19].

To overcome these challenges, reusing foundational and/or core ontologies is recognized as a promising approach [20]. Foundational ontologies such as UFO [21], and DOLCE [22] define a range of top-level domain-independent ontological categories that form a general foundation for more elaborated domain-specific ontologies. Core ontologies such as UFO-L [23], and LKIF-Core [24] in the legal domain provide a precise definition of structural knowledge in a specific field that spans across different domain applications. Ontology reuse can also be accomplished using modeling solutions such as Ontology Patterns (OPs) [25]. OPs describe particular recurring modeling problems that arise in specific ontology development contexts, and present well-proven solutions for the problems [20]. In the legal domain, *part of the design problems can be simplified by creating or extracting conceptual ontology design patterns* [26].

In this paper, which is an extension of a prior introductory work[3], we aim to capture the essentials of GDPR in a *well-founded* legal domain ontology named OPPD (Ontology for the Protection of Personal Data). The concept of "well-founded" ontologies is addressed mainly in Guizzardi's [21] and Guarino's [27] studies. This concept means that ontologies are "faithful to reality" in the sense that *the basic primitives they are built on are sufficiently well-chosen and axiomatized to be generally understood* [27]. To Build OPPD, Ontology-Driven Conceptual Modeling (ODCM), ontology layering, modularization, and reuse processes are applied. ODCM, which is described by *applying ontological analysis based on foundational ontologies to improve the theory and practice of conceptual modeling* [28], is used for grounding OPPD in UFO. Ontology layering and modularization aim to simplify the building process. Ontology reuse focuses on selecting and reusing Conceptual Ontology Patterns (COPs) from UFO and UFO-L. Furthermore, these patterns are applied either by extension or analogy with the legal rules to build the domain content of OPPD. The intention of OPPD is to overcome the lack of a representation of legal procedures that most ontologies encountered. The potential use of OPPD is proposed to formalize GDPR rules by combining ontological reasoning and Logic Programming [29]. The rest of the paper is organized as follows. Section 2 outlines the background of this work. In Section 3, the methodology of building OPPD is presented. Section 4 describes OPPD. The ontology validation and evaluation are discussed in section 5. The ontology potential use is introduced in section 6. Section 7 analyzes the related work. Finally, sections 8 and 9 discuss and conclude the paper respectively.

---

[3] Abstract paper accepted at ICAIL's workshop (2019) - LegRegSW (Legislation and Regulation on the Semantic Web) - Available from: https://sites.google.com/view/legregsw2019/home

2. Background: UFO and UFO-L

This section outlines UFO [21] and UFO-L [23] as our study's background. UFO is a foundational ontology that employs results from formal ontology, cognitive psychology, linguistics, and philosophical logic. It is composed of three main layers: UFO-A [28] (ontology of substances), UFO-B [30] (ontology of events), and UFO-C [28] (ontology of social entities). UFO has been employed in the design of the ontologically well-founded conceptual modeling language OntoUML [21,31]. OntoUML uses the ontological constraints of UFO as modeling primitives and is specified above the UML2.0 meta-model [21]. We referred to UFO as a foundation since it comprises a rich theory of relations and complex relational properties absent in other foundational ontologies [32]. UFO has been successfully applied in many domains ranging from natural science to social domains [33]. Besides, the availability of OntoUML permits the building of ontologies by reusing the generic concepts of UFO as modeling primitives [34].

UFO-L is a legal core ontology developed based on UFO to represent Alexy's theory of fundamental rights [35]. UFO-L defines a variety of basic legal core concepts representing, among many others, *legal roles* (e.g., Right Holder, Duty Holder, etc.), *legal relators* (e.g., Right-Duty Relator, Power-Subjection Relator, etc.), *legal moments* (e.g., Right to an Action, Duty to Act, etc.). Besides, UFO-L specifies a variety of legal patterns representing *legal relators* such as Right-Duty to an Action Relator (Figure 1).

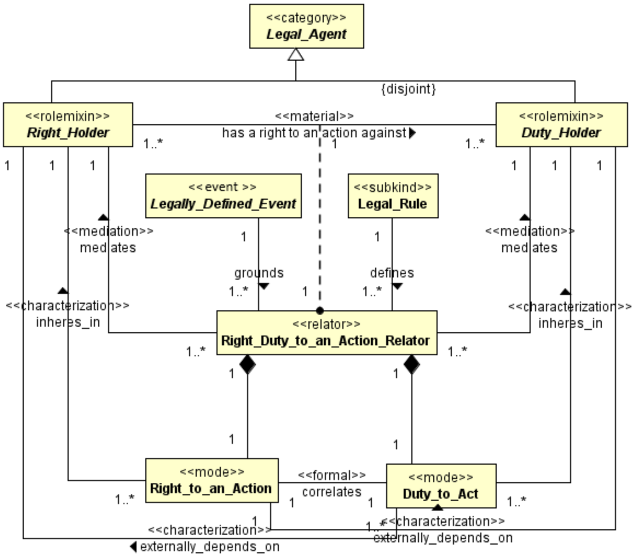


Figure 1. Right_Duty_to_an_Action_Relator represented in OntoUML (adapted from [23]).

The legal patterns aim to represent the legal relations and capture the legal roles played in the context of these relations [36]. In these patterns, legal relators, which are composed of two correlated legal moments, mediate two disjoint legal roles. Each legal moment is inherited in a legal role and externally dependent on the disjoint legal role [23]. In UFO-L, *legal moments* are based on *legal positions* in Alexy's theory. They are defined as situations in which a subject, in a legal relation, for instance, has a right or a power against another subject [34].

3. Methodology

The methodology of building OPPD, inspired by the Systematic Approach for Building Ontologies (SABiO) [25], is composed of five main phases. SABiO differs between *reference* and *operational* ontologies. The former category represents a particular kind of conceptual model developed to make the best possible description of the domain in reality [37]. The latter represents the implementation of reference models as machine-readable artifacts [25].

Ontology Specification OPPD aims to capture the basics of the GDPR, especially the rights of natural persons (named *Data Subject*) to the protection of personal data regarding their processing by organizations (named *Controller* or *Processor*). Besides, the duties and responsibilities of organizations concerning the processing of personal data are also considered. For this purpose, we are referred to a corpus of selected articles and chapters that give rise to issues such as analysis, processing, and interpretation of personal data[4]. The corpus comprises mainly 45 articles that bear on norms. OPPD will be used to model and formalize the legal rules of the GDPR for legal reasoning or compliance checking purposes.

Ontology Requirements They are composed of functional and non-functional. The functional, or *quality* requirements, describe the *goals for modeling* [38]. They are concerned with goals that should be achieved by the modeling process [38] and will be stated as competency questions (CQs) throughout the process. OPPD's functional requirements are: *R1*-To represent the GDPR main agents (e.g., natural persons, controller, public authorities, etc.) and objects (e.g., legal rules). *R2*-To define the essential events and situations (e.g., personal data processing, loss or destruction, infringement of regulation, etc.). *R3*-To determine the basic legal relationships and the active legal roles. Examples of legal relationships between the data subject and the controller are personal data processing, right-duty to rectify or erase data, and right-duty to withdraw consent. *R4*-To describe the legal moments that compose the legal relationships, such as the data subject's right against the controller to rectify his data. Moreover, OPPD has to fulfill the following non-functional requirements concerned with the ontology design: *R1*-The modularity of the ontology to simplify the building process. *R2*-The ontology model needs to provide a clear separation of the structural from the domain knowledge. *R3*-The ontology should be shareable and applicable for building automated applications.

Ontology Design Aiming to simplify the building process of OPPD, we propose ontology layering that divides the ontology structure into three layers located at different granularity levels (see Figure 2). The *upper* layer, located at the most abstract level, contains domain-independent categories (e.g., agent, object, event, situation, etc.). The *core* layer includes categories commonly dependent on the legal domain (e.g., legal agent, legal rule, legal role, etc.). The *domain* layer describes the main categories of the GDPR (e.g., data subject, controller, personal data, consent, etc.). Besides, ontology modularization suggested in SABiO is applied within each layer to facilitate OPPD's building and permit reusing parts of the ontology. Three main ontology modules criteria are considered: independence, coherence and size [25]. Regarding the size, an ontology module aims to cover a sufficient understandable and graphically convenient representation of the problem addressed by this module.

---

[4] Available from: https://sites.google.com/view/legregsw2019/home

Conceptualization To develop the reference model of OPPD, a set of COPs, identified as ontology modules, are selected and reused from UFO and UFO-L to build the upper and core layers. Two main types of patterns are distinguished: *recognition* patterns defined as recurring set of concepts and relations and *template* patterns that describe a common perspective on how to solve a specific problem [39]. The recognition patterns are applied by extension for building the *static*, or *structural*, content of the domain layer. Meanwhile, the *template* patterns are applied by analogy with the legal norms for building the *dynamic*, or *procedural*, content. This phase is performed using ODCM to ground OPPD in UFO. Thus, the ontologically-founded conceptual modeling language OntoUML [21,31] is applied to represent the conceptual patterns and their application using the modeling primitives of UFO. Thereby, the OntoUML constraints for relating these primitives are respected (see Figure 2 for an example).

Ontology Validation and Evaluation This phase consists of (1) validating OPPD by transforming the reference model into an *operational* ontology represented using a computational language and (2) evaluating the ontology against the requirements.

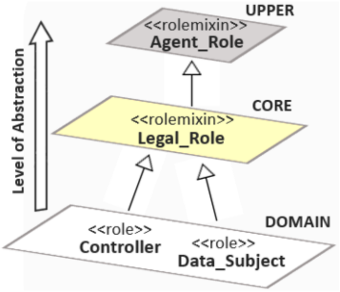


Figure 2. The layered structure of OPPD.

4. OPPD: The Reference Ontology Model

4.1. Upper and Core Layers

This section presents briefly part of the upper and core layers due to space limitation. For building the upper layer, three main COPs, considered as recognition patterns, are selected from UFO: Substance, Event and Moment.

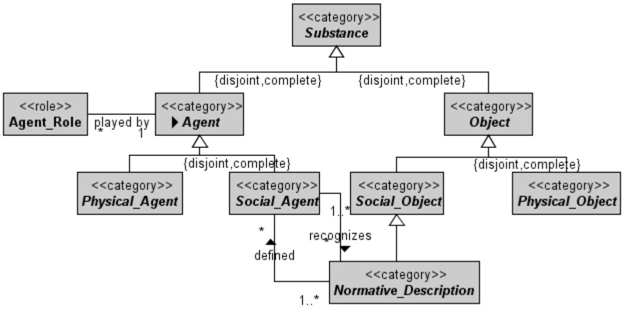


Figure 3. Substance ontology pattern (adapted from [28]).

In Figure 3, the Substance ontology pattern, which implements the functional requirement R1, is depicted. In this pattern, Substance can be Agent or Object. Normative_Description is a Social_Object recognized by at least one Social_Agent and Agent_Role is played by one Agent.
For building the core layer, different COPs are selected from UFO-L [23]. The recognition patterns are: Legal_Substance, Legal_Event, Externally_Dependent_Legal_Moment, and Legal_Relator. The template patterns are: Right_Duty_to_an_Action_Relator (Figure 1), Right_Duty_to_Omission_Relator, and Power_Subjection_Relator [23]. In Figure 4, Externally_Dependent_Legal_Moment, which implements the functional requirement R4, is depicted. In this pattern, different legal moments are considered [36]: Right (i.e., legal moment in which one may demand from another the performance of a certain conduct), Right_to_an_Action, Duty (converse moment of Right), Duty_to_Act, Legal_Power (i.e., ability to act to a power holder), Legal_Subjection (converse moment of Legal_Power), Disability (i.e., no power to create, change or extinguish a legal moment), and Immunity (converse moment of Disability).

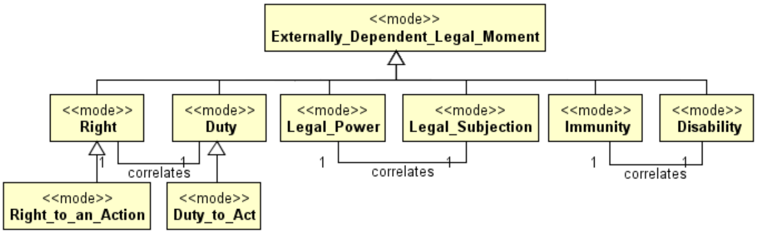


Figure 4. Externally_Dependent_Legal_Moment ontology pattern (adapted from [23]).

4.2. Domain Layer

This section addresses OPPD's domain layer developed by applying the core layer's patterns. The domain layer's static content is obtained by extending the core layer's recognition patterns. For this purpose, a list of competency questions (CQs) is outlined to fulfill the functional requirements R1 and R2, such as: (CQ1) What are the leading legal agents defined in the GDPR? (CQ2) What are the main legal objects? (CQ3) What are the critical legal events? (CQ4) What are the primary legal situations? (CQ5) What are the main legal relators? (CQ6) What are the essential legal moments? Examples of extending core layer's concepts are illustrated in Figures 5, 6, and 7. Concerning the procedural content, the core layer's template patterns are applied by analogy with the norms of the GDPR. A list of CQs is addressed for each pattern, which is considered ontology module or sub-ontology, to fulfill the functional requirements R3 and R4.

Application of Right_Duty_to_an_Action_Relator This legal relator represents the relationship where the *Right_Holder has the right to a positive action by the Duty_Holder* [23]. The relator pattern can be applied by analogy with several legal rules such as Art. 7 (Right_Duty_to_Withdraw_Consent), Art. 15 (Right_Duty_to_Processing_Confirmation), Art. 16 (Right_Duty_to_Rectification), and Art. 82 (Right_Duty_to_Compensation).

> *Article 7. Section 3. The data subject shall have the right to withdraw his or her consent at any times [...]*

> *Article 15. Section 1. The data subject shall have the right to obtain from the controller confirmation as to whether or not personal data are being processed [...]*

> *Article 16. The data subject shall have the right to obtain from the controller the rectification of inaccurate personal data [...]*

> *Article 82. Section 1. Any person who has suffered material or non-material damage as a result of an infringement of this Regulation shall have the right to receive compensation from the controller or processor for the damage suffered.*

Figure 5 presents an ontology module that defines Right_Duty_to_Compensation, realized by applying Right_Duty_to_Action_Relator by analogy with Art. 82. The following CQs are addressed for applying this pattern: (CQ1) What GDPR legal rule has been infringed? (CQ2) Who is involved in the GDPR infringement? (CQ3) What damage has resulted from the GDPR infringement? (CQ4) What personal data was affected by this damage? (CQ5) Who do the personal data identify as legal agent? (CQ6) Who is suffering from the personal data damage? (CQ7) Who has the right to compensation? (CQ8) Who is charged by the compensation?

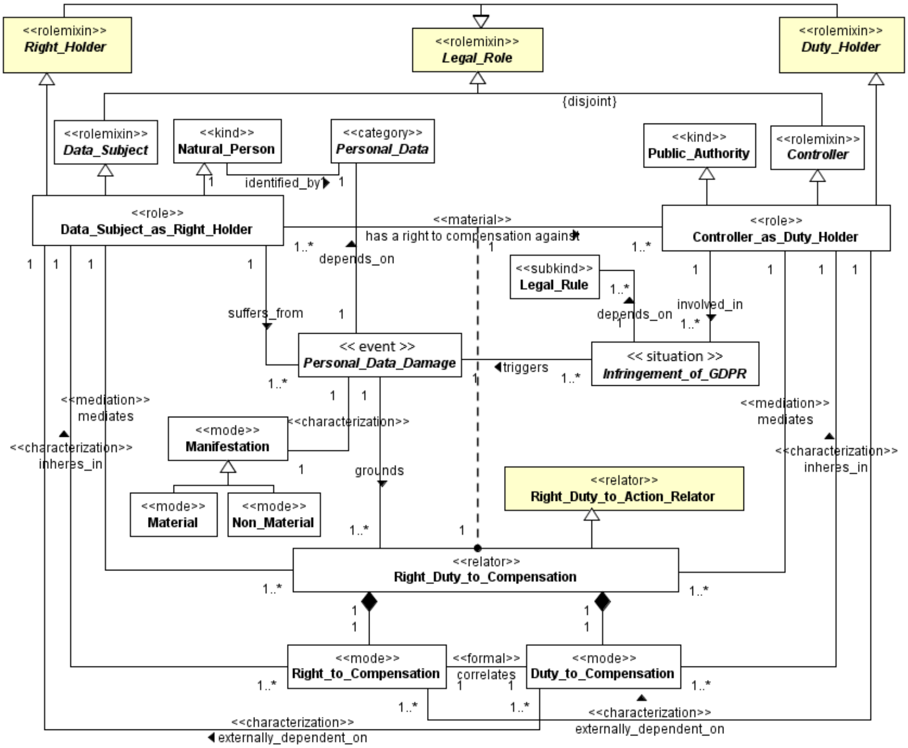


Figure 5. Right_Duty_to_Compensation represented in OntoUML.

Application of Right_Duty_to_Omission_Relator This legal relator represents the relationship where the *Right_Holder has the right to an omissive duty action by the Duty_Holder*. In other words, the Duty_Holder has a duty to refrain from acting [23]. Figure 6 depicts an ontology module, that defines Right_Duty_to_Objection_to_Processing, achieved by applying Right_Duty_to_Omission_Relator by analogy with Art. 21. The following CQs are addressed for applying this pattern: (CQ1) For what purposes personal data processing is performed? (CQ2) On what personal data is the processing dependent? (CQ3) Who do the personal data identify? (CQ4) Who is the legal agent suffering from the processing? (CQ5) Who is involved in the personal data processing? (CQ6) Who has the right to object to personal data processing? (CQ7) Who is charged by terminating the personal data processing?

> *Article 21. Section 2. Where personal data are processed for direct marketing purposes, the data subject shall have the right to object at any time to processing [...]*

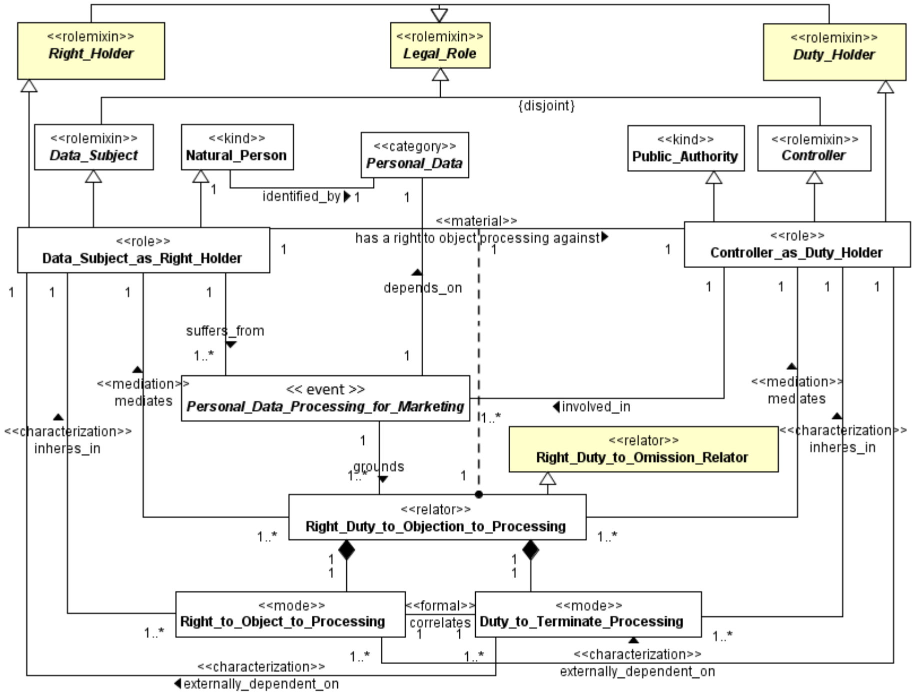


Figure 6. Right_Duty_to_Objection_to_Processing represented in OntoUML.

Application of Power_Subjection_Relator This legal relator represents the relationship where the *Power_Holder has the competence (or the legal power) to create (change, extinguish) a legal position or a situation against the Subjection_Holder* [23]. In Figure 7, we present an ontology module, that defines Power_Subjection_to_Liabilities, obtained by applying Power_Subjection_Relator by analogy with Art. 82. The following CQs are addressed for applying this pattern: (CQ1) What GDPR legal rule has been infringed by personal data processing? (CQ2) On what personal data is the processing dependent? (CQ3) Who is involved in the personal data processing? (CQ4) What damage is caused by the GDPR infringement? (CQ5) Who is suffering from the personal data damage resulting from the infringement? (CQ6) Who has the power to define liability towards the damage? (CQ7) Who is the liable legal agent for the personal data damage? (CQ8) What legal relator was created based on the power subjection relationship?

Article 82. Section 2. Any controller involved in processing shall be liable for the damage caused by processing which infringes this Regulation [...]

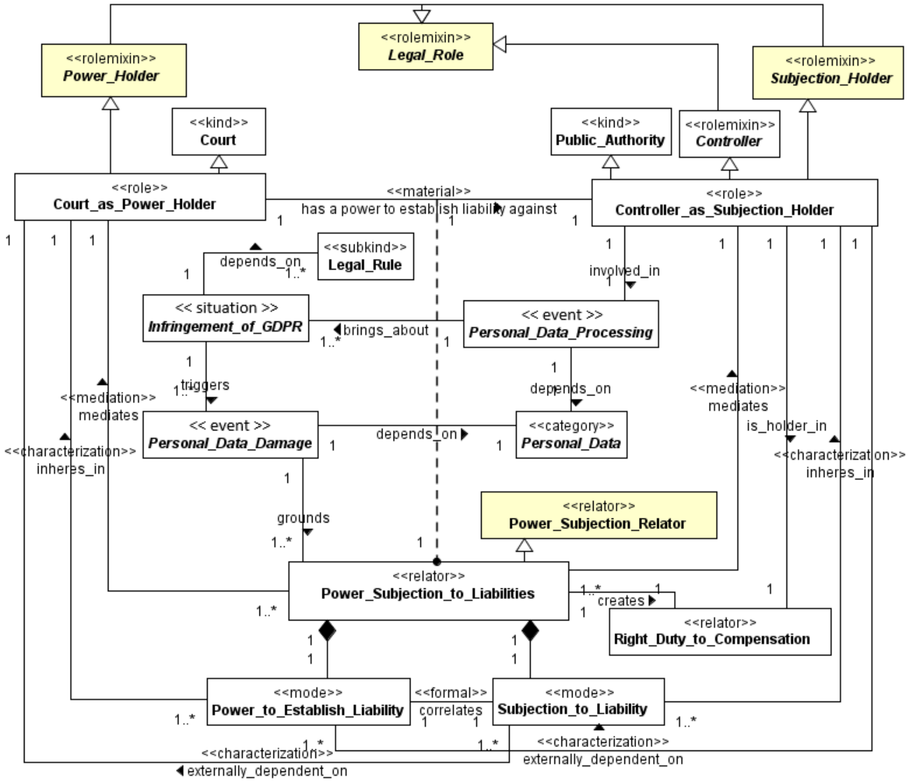


Figure 7. Power_Subjection_to_Liabilities represented in OntoUML.

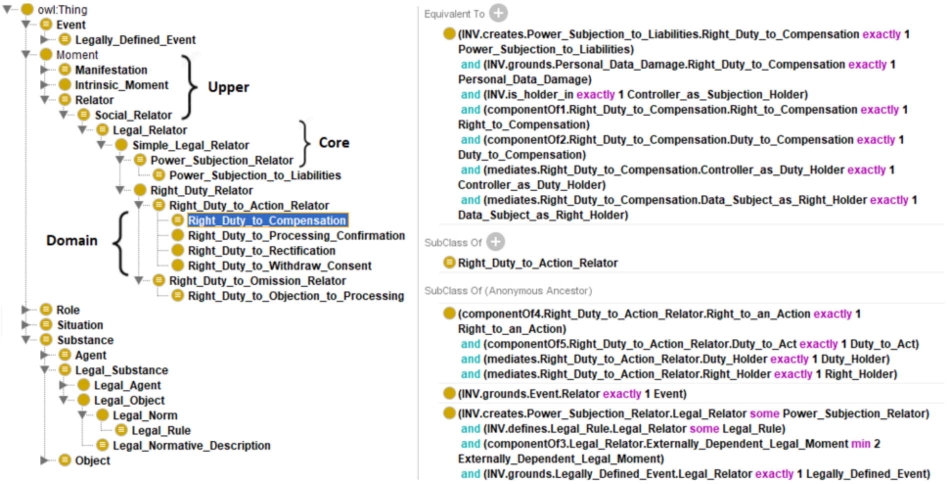
5. Ontology Validation and Evaluation

In this section, we validate OPPD by transforming the reference model into an operational ontology. The ontology environment OLED [40] provides the transformation by generating the OWL code. The code generator maps OntoUML classes, associations, and attributes to OWL classes, object properties, and data properties. It considers generalization sets and their disjointness properties plus model cardinalities. The code generator transforms to SWRL[5] rules the domain constraints and cardinalities and the transitivity of material and parthood relations. The resulted operational ontology is manageable in ontology editors such as Protégé (see Figure 8 for an example). OPPD's consistency is verified using HermiT, an OWL2 inference engine[6]. The ontology, under construction, contains 89 classes, 83 subClassOf relations, 56 equivalent classes, 234 object properties, and 14 disjoint axioms.

---

[5] https://www.w3.org/Submission/SWRL/

[6] http://www.hermit-reasoner.com/



```
(INV.creates.Power_Subjection_to_Liabilities.Right_Duty_to_Compensation exactly 1
Power_Subjection_to_Liabilities)
 and (INV.grounds.Personal_Data_Damage.Right_Duty_to_Compensation exactly 1
Personal_Data_Damage)
 and (INV.is_holder_in exactly 1 Controller_as_Subjection_Holder)
 and (componentOf1.Right_Duty_to_Compensation.Right_to_Compensation exactly 1
Right_to_Compensation)
 and (componentOf2.Right_Duty_to_Compensation.Duty_to_Compensation exactly 1
Duty_to_Compensation)
 and (mediates.Right_Duty_to_Compensation.Controller_as_Duty_Holder exactly 1
Controller_as_Duty_Holder)
 and (mediates.Right_Duty_to_Compensation.Data_Subject_as_Right_Holder exactly 1
Data_Subject_as_Right_Holder)

Right_Duty_to_Action_Relator

(componentOf4.Right_Duty_to_Action_Relator.Right_to_an_Action exactly 1
Right_to_an_Action)
 and (componentOf5.Right_Duty_to_Action_Relator.Duty_to_Act exactly 1 Duty_to_Act)
 and (mediates.Right_Duty_to_Action_Relator.Duty_Holder exactly 1 Duty_Holder)
 and (mediates.Right_Duty_to_Action_Relator.Right_Holder exactly 1 Right_Holder)
(INV.grounds.Event.Relator exactly 1 Event)
(INV.creates.Power_Subjection_Relator.Legal_Relator some Power_Subjection_Relator)
 and (INV.defines.Legal_Rule.Legal_Relator some Legal_Rule)
 and (componentOf3.Legal_Relator.Externally_Dependent_Legal_Moment min 2
Externally_Dependent_Legal_Moment)
 and (INV.grounds.Legally_Defined_Event.Legal_Relator exactly 1 Legally_Defined_Event)
```

Figure 8. Excerpt of OPPD (Right_Duty_to_Compensation) represented in Protégé.

Moreover, to evaluate OPPD's validity regarding its requirements, a set of SPARQL[7] queries are formalized to translate the competency questions defined in the conceptualization phase. The queries are executed to interrogate OPPD's concepts and relations using defined instances. In the following, excerpt of the SPARQL queries translated from the CQs specified for modeling Right_Duty_to_Compensation (Figure 5) is presented:

```
(CQ1) SELECT ?rule WHERE {?rule rdf:type oppd:Legal_Rule.
?infringement rdf:type oppd:Infringement_of_GDPR.
?infringement oppd:depends_on ?rule.}
(CQ2) SELECT ?controller WHERE {?controller rdf:type oppd:Controller.
?infringement rdf:type oppd:Infringement_of_GDPR.
?controller oppd:involved_in ?infringement.}
(CQ3) SELECT ?damage WHERE {?infrigement rdf:type oppd:Infringement_of_GDPR.
?infrigement oppd:triggers ?damage.}
(CQ4) SELECT ?personaldata WHERE {?damage rdf:type oppd:Personal_Data_Damage.
?damage oppd:depends_on ?personaldata.}
```

6. Ontology Use

OPPD intends to be helpful for different purposes such as data querying, information retrieval, legal reasoning, and compliance checking. In this study, we present a preliminary work that demonstrates the potential use of OPPD to formalize the GDPR legal rules for reasoning practices. Rule language such as SWRL can be used for the formalization process (e.g., [41]) yet limit the reasoning purposes since it lacks the non-monotonic features. The objective of non-monotonic reasoning is to develop reasoning systems that model how common sense is used by humans [42]. As *non-monotonic reasoning is related to Logic Programming* [43] - in the sense that they share common goals and techniques (e.g., negation as failure) - we envisaged this approach to achieve our intention to combine logic programs with ontological reasoning. Using SWI-Prolog [44] and Thea[8],

---

[7] https://www.w3.org/TR/sparql11-query/

[8] Prolog library for managing OWL2 ontologies. Available from: https://github.com/vangelisv/thea

OPPD, represented as OWL abstract syntax terms, is converted into a logic program. The mapping process implements Description Logic Programs (DLP) [45]. For instance, OPPD's axioms (A) are converted to the Prolog program (B).

(A)

```
subClassOf(Data_Subject,Legal_Role)
classAssertion(Legal_Rule,Article_82)
inverseProperties(INV.mediates,mediates)
```

(B)

```
Legal_Role(X) :- Data_Subject(X).
Legal_Rule(Article_82).
INV.mediates(X,Y) :- mediates(Y,X).
mediates(X,Y) :- INV.mediates(Y,X).
```

Furthermore, the generated logic program will be used for encoding logic rules representing the procedural aspect of GDPR legal rules. For instance, the subsequent set of rules is proposed to represent Art. 82.1. (Figure 5). In this context, the isomorphism principle, stated by Bench-Capon [46], is followed to create a *well-defined correspondence* between the rules in the formal model and the rules in natural language.

```
(1) Right_Duty_to_Compensation(Z) :- Controller(X), Data_Subject(Y),
Personal_Data(P), Infringement_of_GDPR(I), involved_in(X,I),
Personal_Data_Damage(D), triggers(I,D), suffers_from(Y,D), depends_on(D,P),
Legal_Relator(Z), grounds(D,Z).
(2) mediates(Z,X) :- Right_Duty_to_Compensation(Z), Controller(X).
(3) mediates(Z,Y) :- Right_Duty_to_Compensation(Z), Data_Subject(Y).
(4) has_a_right_to_compensation_against(Y,X) :- Right_Duty_to_Compensation(Z),
Controller(X), Data_Subject(Y), mediates(Z,X), mediates(Z,Y).
```

7. Related Work

As related to this study, two main preferences are defined, the legal core ontologies and the domain ontologies representing the GDPR norms. Therefore, LKIF-Core [24] is selected being the latest validated legal core ontology preceding UFO-L. Besides, two domain ontologies are considered, PrOnto [12] as a legal domain ontology covering the GDPR norms and concepts and GConsent that represents the consent concept based on the GDPR [11].

LKIF-Core [24] is a legal core ontology implemented in OWL[9]. It is composed of 15 modules categorized into three main categories, each of which describes a set of closely related concepts from both legal and commonsense domains: (i) *Abstract Concepts* define the place, mereology, time, and spacetime; (ii) *Basic Concepts* define concepts around process, role, action, and expression; (iii) *Legal Concepts* define legal action, legal role, and norm. LKIF-Core considers the representation of *normative knowledge* which is the basis of the normative reasoning in the AI and Law research domain. It provides interpretations for the terms *obligation*, *prohibition* and *permission* [47]. By reusing LKIF-Core, the representation of the hierarchical structure of basic legal concepts is maintained. However, the main drawback is the lack of interpretation of the procedural aspect of provisions. This deficiency is admitted in the literature that *most ontologies did not have an adequate solution for legal procedures mainly because of the difficulty to find a language to express knowledge in a declarative way* [48].

---

[9] https://github.com/RinkeHoekstra/lkif-core

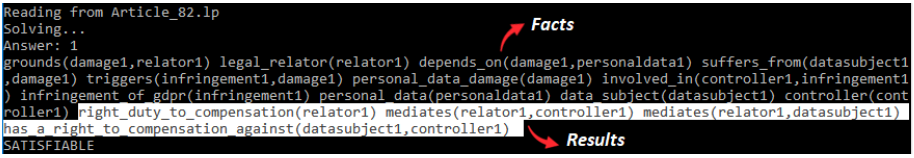
PrOnto [12] is a legal ontology developed for modeling the GDPR concepts and norms. Its main goal is to support legal reasoning and compliance checking by employing defeasible logic theory. PrOnto is developed using the MeLOn methodology, which iterates over ten steps. To resume, we outline five essential steps that are commonly addressed in ontology engineering approaches (as described by the authors in [12]): Describe the ontology goal; Reuse existing ontologies, design patterns, or domain vocabularies; Use usable tools (e.g., tables, UML diagrams, and the Graffoo tool); Refine and optimize the ontology with the help of an ontology expert that manually adds the axioms; Ontology evaluation. In PrOnto, concepts such as Agent and Role and relations such as *plays* are reused from LKIF-Core to represent legal roles (e.g., Controller) aiming to model obligations and rights in the GDPR (e.g., Right to Data Portability). PrOnto is composed of different modules: (i) data and documents, (ii) agents and roles, (iii) processing purposes and legal bases; (iv) data processing and workflow, risk management, and (v) legal rules and deontic operators. GConsent [11] is an OWL2-DL ontology for representing information associated with consent, specifically, the given aspect of consent i.e., consent provided by the data subject. For building GConsent, the “Ontology Development 101” methodology [49] is used by applying the following phases: Gather information about consent from GDPR, articles, academic papers; Create use-cases and competency questions based on collected information; Create ontology to express information about use-cases; Ontology evaluation. The core concepts defined in GConsent are Consent, Data Subject, Personal Data, Purpose, Processing, and Status. To conclude, in PrOnto and GConsent, the representation of legal concepts and relations is established. However, an explicit description of legal procedures is not supported. The legal relations and the active legal roles are not represented in a procedural perspective required to describe the procedural aspect of the GDPR norms.

8. Discussion

This study’s main contribution is developing a well-founded legal domain ontology, named OPPD, representing the essentials of the GDPR. For building OPPD, a pattern-oriented approach is applied, supported by ontology grounding, layering, modularization, and reuse processes. Conceptual Ontology Patterns are selected from the foundational ontology UFO and the legal core ontology UFO-L. These patterns are reused as ontology modules and applied either by extension or analogy with legal rules to build OPPD’s domain content. Besides, the ontology-driven conceptual modeling process (ODCM) is used for grounding OPPD in UFO. In this approach, we differentiate between the ontology’s reference and operational versions. The reference model of OPPD, which is independent of any computational language, is implemented in OWL and SWRL. Besides, the structural and domain knowledge in OPPD are separated. The former represents the hierarchy of concepts (e.g., Figures 3 and 4) which is distinguished from modeling the legal rules’ procedural aspects (e.g., Figures 5 and 7). This distinction will support the extensibility of the ontology to include future aspects. The second contribution is that by reusing UFO-L’s legal relators’ patterns and applying them by analogy with the legal rules, we obtained a richly populated ontology representing the procedural aspects of these rules. This result is considerable for employing our ontology for reasoning or compliance checking purposes and is difficult to achieve by reusing other legal core ontologies that lack the representation of legal procedures. This deficiency is admitted in the literature that most approaches have failed to support the representation of the legal relations and capture the legal roles played in the context of these relations [36]. Finally, this study has two main limitations. First, the personal data processing purposes are not considered. For instance, in Figure 6, *marketing purpose* is defined jointly with the event Personal_Data_Processing_for_Marketing. However, there is a need to distinguish the event from the event's purpose. Second, due to Prolog syntax that prevents conjunctions and disjunctions in the rule's head, a single legal rule (e.g., Art. 82.1.) is formalized using multiple logic rules (see Section 6 for an example), which may affect a more beneficial application of the isomorphism principle.

9. Conclusion

Legal ontologies are considered the missing link between legal theory and AI & law. They provide stable foundations for knowledge representation in the legal domain. However, their development is challenging due to the complexity of the legal knowledge. This study demonstrated that applying a pattern-based approach supported by ontology reuse, modularization, and layering processes is a practical strategy to overcome the existing difficulties. The support processes aimed to simplify the ontology development by reusing conceptual patterns from the foundational ontology UFO and the legal core ontology UFO-L. Besides, ODCM is used for grounding OPPD in UFO. As a result, we obtained OPPD, a well-founded legal domain ontology with a significant ontological expressiveness. OPPD is validated by implementing the ontology in OWL and evaluated using SPARQL queries translated from the defined CQs. Finally, preliminary work to formalize GDPR legal rules by integrating OPPD and Prolog is presented. In further works, we will proceed with the ontology development to address other legal aspects such as consent, immunities to liabilities, and processing purposes. Furthermore, OPPD's *semantic accuracy* will be assessed by computing different structural measures (e.g., depth, average depth, depth variance, etc.) [50]. Concerning the reasoning over the formalized logic rules, it will be maintained using Answer Set Programming (ASP) [51]. ASP solvers binding to SWI-Prolog are required to solve logic programs (see Figure 9).



```
Reading from Article_82.lp
Solving...
Answer: 1
grounds(damage1,relator1) legal_relator(relator1) depends_on(damage1,personaldata1) suffers_from(datasubject1
,damage1) triggers(infringement1,damage1) personal_data_damage(damage1) involved_in(controller1,infringement1
) infringement_of_gdpr(infringement1) personal_data(personaldata1) data_subject(datasubject1) controller(cont
roller1) right_duty_to_compensation(relator1) mediates(relator1,controller1) mediates(relator1,datasubject1)
has_a_right_to_compensation_against(datasubject1,controller1)
SATISFIABLE
```

Figure 9. An example of solving the logic program representing Art. 82.1.

References

[1] EU-Commission. General Data Protection Regulation. Official Journal of the European Communities. 2018.

[2] Teixeira GA, Silva M, Pereira R. The critical success factors of GDPR implementation: a systematic literature review. Digital Policy, Regulation and Governance. 2019;21(4):402-418.

[3] Romanou A. The necessity of the implementation of Privacy by Design in sectors where data protection concerns arise, Computer Law & Security Review. 2018;34:99-110.

[4] Lopes IM, Guarda T, Oliveira P. EU General Data Protection Regulation Implementation: An Institutional Theory View. Proceedings of WorldCIST'19, New Knowledge in Information Systems and Technologies. Advances in Intelligent Systems and Computing; vol. 930. Springer, Cham; 2019. p. 383-393.

[5] Tikkinen-Piri C, Rohunen A, Markkula J. EU General Data Protection Regulation: Changes and implications for personal data collecting companies. Computer Law and Security Review. 2018;34(1):134-153.

[6] Kingston, J. Using artificial intelligence to support compliance with the general data protection regulation. Artificial Intelligence and Law. 2017;25:429-443. doi: 10.1007/s10506-017-9206-9.

[7] Torre D. An AI-assisted Approach for Checking the Completeness of Privacy Policies Against GDPR. Proceedings of IEEE 28th International Requirements Engineering Conference (RE), Zurich, Switzerland; 2020. p. 136-146; doi: 10.1109/RE48521.2020.00025.

[8] Brkan M. AI-supported decision-making under the general data protection regulation. Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law; 2017. p. 3-8.

[9] Torre D, Soltana G, Sabetzadeh M, Briand LC, Auffinger Y, Goes P. Using Models to Enable Compliance Checking Against the GDPR: An Experience Report. Proceedings of ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS); Munich, Germany; 2019. p. 1-11. doi: 10.1109/MODELS.2019.00-20.

[10] Zeni N, Kiyavitskaya N, Mich L, Cordy J, Mylopoulos J. GaiusT: supporting the extraction of rights and obligations for regulatory compliance. Requirements Eng. 2015;20:1–22.

[11] Pandit HJ, Debruyne C, O'Sullivan D, Lewis D. GConsent - A Consent Ontology based on the GDPR. Proceedings of 16th European Semantic Web Conference (ESWC); 2019.

[12] Palmirani M, Martoni M, Rossi A, Bartolini C, Robaldo L. Legal Ontology for Modelling GDPR Concepts and Norms. In Monica Palmirani, editor, Legal Knowledge and Information Systems - JURIX 2018: The Thirty-first Annual Conference, Groningen, The Netherlands; 2018 December 12-14; IOS Press; 2018. p. 91-100.

[13] Bonatti PA, Ioffredo L, Petrova IM, Sauro L, Siahaan I. Real-time reasoning in OWL2 for GDPR compliance. Artificial Intelligence. 2020 jan;289. doi:arxiv-2001.05390.

[14] Valente A, and Breuker J. Ontologies: The missing link between legal theory and AI & law. In: H. Prakken, A.J. Muntjewerff, and A. Soeteman, editors. Proceedings of Legal Knowledge Based Systems Jurix 1994; Lelystad: Vermande; 1994.

[15] Mommers L. Application of a knowledge-based ontology of the legal domain in collaborative workspaces. Proceedings of the 9th international conference on Artificial intelligence and law; 2003. p. 70-76.

[16] Benjamins VR, Casanovas P, Breuker J, Gangemi A. Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval and Applications. Springer; 2005.

[17] Francesconi E, Tiscornia D. Building Semantic Resources for Legislative Drafting: The DALOS Project. Computable Models of the Law. LNCS; Springer-Verlag Berlin, Heidelberg; 2008. p. 56-70.

[18] Dove I. Legal Expert Systems: The End of Jurisprudence?. The Journal of Legal Studies in Business. 1996.

[19] Gardner A. Book Review. An Artificial Intelligence Approach To Legal Reasoning. MIT Press; 1987. p. 223-233.

[20] Falbo R, Guizzardi G, Gangemi A, Presutti V. Ontology Patterns: Clarifying Concepts and Terminology. Proceedings of WOP; 2013. p. 14-26.

[21] Guizzardi G. Ontological Foundations for Structural Conceptual Models. Telematica-Institut/CTIT. 2005.

[22] Borgo S, Masolo C. Ontological foundations of DOLCE; Theory and Applications of Ontology: Computer Applications. Springer; 2010.

[23] Griffo C. UFO-L, A Core Ontology of Legal Aspects Building Under the Perspective of Legal Relations. Federal University of Espirito Santo. 2018.

[24] Hoekstra R, Breuker J, Di Bello M, Boer A. LKIF Core: Principled Ontology Development for the Legal Domain. Proceedings of the 2009 conference on Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood; IOS Press; 2009. p. 21-52.

[25] Falbo R. SABiO: Systematic Approach for Building Ontologies. 2014.

[26] Gangemi A. Design Patterns for Legal Ontology Constructions. Proceedings of the 2nd Workshop on

Legal Ontologies and Artificial Intelligence Techniques; USA; 2007.
[27] Guarino N, Oberle D, Staab S. What is an Ontology? In: Staab S, Studer R, editors. Handbook on Ontologies. International Handbooks on Information Systems. Springer, Berlin, Heidelberg; 2009. doi: 10.1007/978-3-540-92673-3_0
[28] Guizzardi G, Falbo R, Guizzardi RS. Grounding software domain ontologies in the unified foundational ontology (UFO): The case of the ode software process ontology. Proceedings of the 1th Iberoamerican Workshop on Requirements Engineering and Software Environments; 2008.
[29] Kowalski R. Predicate logic as a programming language. Proceedings of IFIP Congress 74; 1974; p. 569–574.
[30] Guizzardi G. Towards Ontological Foundations for the Conceptual Modeling of Events. Proceedings of ER 2013; Springer, Heidelberg; 2013. p. 327-341.
[31] Guizzardi G, Fonseca CM, Benevides AB, Almeida JPA, Porello D, Sales TP. Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0. Proceedings of ER 2018; 2018.
[32] Guizzardi G, Wagner G, Falbo R, Guizzardi R S. Towards Ontological Foundations for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story. Applied Ontology. 2015;10(3-4):259-271.
[33] Falbo R, Baiao F, Lopes M, Guizzardi G. The Role of Foundational Ontologies for Domain Ontology Engineering: An Industrial Case Study in the Domain of Oil and Gas Exploration and Production. International Journal of Information System Modeling and Design. 2010;1:1-22.
[34] Griffo C, Almeida J, Guizzardi G. Towards A Legal Core Ontology Based On Alexy's Theory of Fundamental Rights. Proceedings of ICAIL Multi-Lingual Workshop on AI and Law (MWAIL); 2015.
[35] Alexy R. A Theory of Constitutional Rights. Oxford University Press; 2009.
[36] Griffo C, Almeida J, Guizzardi G. Conceptual Modeling of Legal Relations. Proceedings of ER 2018, Conceptual Modeling; 2018. p. 169-183.
[37] Guizzardi G. On Ontology, ontologies, Conceptualizations, Modeling Languages and (Meta)Models. Proceedings of Databases and Information systems; 2007. p. 18-39.
[38] Guizzardi G, Proper HA. On Understanding the Value of Domain Modeling. Proceedings of 15th International Workshop on Value Modelling and Business Ontologies (VMBO 2021); 2021.
[39] Gangemi A, Presutti V. Ontology Design Patterns. Handbook on Ontologies; Springer, Heidelberg. 2009.
[40] Guerson J, Sales TP, Guizzardi G, Almeida JPA. OntoUML Lightweight Editor: A Model-Based Environment to Build, Evaluate and Implement Reference Ontologies. Proceedings of 19th IEEE EDOC; 2015.
[41] El Ghosh M, Abdulrab H. Ontology-Based Liability Decision Support in the International Maritime Law. Proceedings of the 33rd International Conference on Legal Knowledge and Information Systems (JURIX); 2020. p. 273-276.
[42] Minker J. An overview of nonmonotonic reasoning and logic programming. The Journal of Logic Programming. 1993;17(2-4):95-126.
[43] Gelfond M, Leone N. Logic programming and knowledge representation—The A-Prolog perspective. Artificial Intelligence. 2002;138:3–38.
[44] Wielemaker J, Schrijvers T, Triska M, Lager T. SWI-Prolog. Theory and Practice of Logic Programming. 2010;12(1-2):67-96.
[45] Grosof BN, Horrocks I, Volz R, Decker S. Description logic programs: combining logic programs with description logic. Proceedings of the 12th international conference on World Wide Web; 2003. p. 48–57.
[46] Bench-Capon TJM, Coenen FP. Isomorphism and legal knowledge based systems. Artificial Intelligence and Law. 1992;1:65–86. doi: 10.1007/BF00118479.
[47] Breuker J, Valente A, Wikels R. Legal Ontologies in Knowledge Engineering and Information Management. Artificial Intelligence and Law. 2004;12:241-277.
[48] Visser PR, Bench-Capon TJ. A comparison of two legal ontologies. Working papers of the First International Workshop on Legal Ontologies; University of Melbourne, Australia. 1997.
[49] Noy NF, McGuinness DL. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880; 2001.
[50] Sanchez D, Batet M, Martinez M, Domingo-Ferrer J. Semantic variance: An intuitive measure for ontology accuracy evaluation. Engineering Applications of Artificial Intelligence. 2015;39:89-99.
[51] Erdem E, Gelfond M, Leone N. Applications of Answer Set Programming. AI Magazine. 2016;37(3):53-68. doi: 10.1609/aimag.v37i3.2678.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

doi:10.3233/FAIA210379

Foundations of the Socio-Physical Model of Activities (SOMA) for Autonomous Robotic Agents[1]

Daniel Beßler [a], Robert Porzel [b], Mihai Pomarlan [c], Abhijit Vyas [a], Sebastian Höffner [b], Michael Beetz [a], Rainer Malaka [b], and John Bateman [c]
[a] *Department of Artificial Intelligence, University of Bremen, 28359 Bremen, Germany*
[b] *Digital Media Lab, TZI, University of Bremen, 28359 Bremen, Germany*
[c] *Department of Linguistics, University of Bremen, 28359 Bremen, Germany*

Abstract. In this paper, we present foundations of the Socio-physical Model of Activities (SOMA). SOMA represents both the physical as well as the social context of everyday activities. Such tasks seem to be trivial for humans, however, they pose severe problems for artificial agents. For starters, a natural language command requesting something will leave many pieces of information necessary for performing the task unspecified. Humans can solve such problems fast as we reduce the search space by recourse to prior knowledge such as a connected collection of plans that describe how certain goals can be achieved at various levels of abstraction. Rather than enumerating fine-grained physical contexts SOMA sets out to include socially constructed knowledge about the functions of actions to achieve a variety of goals or the roles objects can play in a given situation. As the human cognition system is capable of generalizing experiences into abstract knowledge pieces applicable to novel situations, we argue that both physical and social context need be modeled to tackle these challenges in a general manner. The central contribution of this work, therefore, lies in a comprehensive model connecting physical and social entities, that enables flexibility of executions by the robotic agents via symbolic reasoning with the model. This is, by and large, facilitated by the link between the physical and social context in SOMA where relationships are established between occurrences and generalizations of them, which has been demonstrated in several use cases in the domain of everyday activites that validate SOMA.

Keywords. design patterns, domain ontology, autonomous robotics

1. Introduction

In spite of undoubtedly being ubiquitous, the domain of *everyday activities* poses considerable challenges. Many people perform activities such as cooking or cleaning almost every day. This includes to select and manipulate ingredients, use tools and

---

[1] This work was funded by the German Research Foundation (DFG) as part of Collaborative Research Center (CRC) 1320 EASE – Everyday Activity Science and Engineering, University of Bremen (`http://www.ease-crc.org/`), subprojects H2, P1 and R1.

devices, arrange the prepared dishes for serving and clean up afterwards – and do it all quick and robust without recourse to an advanced computational theory. Further, the amount of information provided in a description of a task – such as a natural language command requesting its completion – is much less than the amount of information needed to perform the task. This raises the question how humans are able to decide so quickly what to do next, despite ambiguity and underspecification.

Lenat and Feigenbaum observe that "more knowledge implies less search" [1]. Knowledge of many possible plans, as well as knowledge of the world in general, seems to be the secret of human performance. There is no algorithmic reason why tomatoes and oregano go well together, or why a raw egg must be handled with care. The cook simply has to know these things. Such knowledge of the world is taught, observed, and then ingrained by practice. As Anderson observes, "an agent has a great deal of knowledge [of everyday activities], which comes as a result of the activity being common" [2]. As human beings, we acquire such knowledge naturally over the course of our lives. A lot of what we learn, we learn by doing, or by watching others. This suggests that a robot must have mechanisms to organize and interpret observations, either of its own behavior or of other agents, into structures that are then amenable for other computational tasks. To this end, the concept of *narratively-enabled episodic memories* (NEEMs) was introduced [3,4]. NEEMs are comprehensive logs of raw sensor data, actuator control histories and perception events, all semantically annotated with information about what the robot is doing and why using the terminology provided by SOMA.

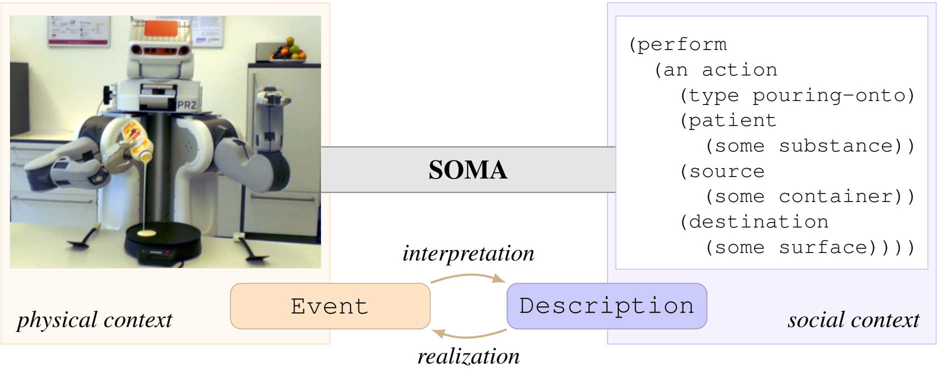
The computational tasks that must be solved when acting in the physical world are often very complex and beyond what is thought to be tractable. This, however, is only the case when these problems are regarded in their full generality and not for restricted versions of these problems. However, the knowledge representing such pragmatic solutions goes beyond modeling physical events and requires models of the social context by means of which the physical events can be realized and interpreted. For this, we employ an existing upper-level ontology that we augment with general design patterns and specific modules that are pertinent for robot knowledge modeling. In this paper we provide an overview, of this approach for robot knowledge modeling where all of our extensions to the given foundational framework rely on the differentiation between the observable physical domain and the conceptualized social interpretations thereof.

The overall goal of our research is to enable robotic agents to perform everyday activities with similar robustness and flexibility as human agents do. Given this aim, we must, in some sense, get the robot to know what humans know about the world, at least as it pertains to everyday activities. This presents several challenges, beyond the scope of what needs to be known to represent such intricate and extensive domain. There lies the question of how to represent and structure this knowledge in order to realize a similar robustness, flexibility and efficiency in performance. In addition, there are challenges concerning the acquisition and learnability of the corresponding structures. In this paper, we will focus on how this knowledge is represented. We will describe our employment of an existing upper level ontology and the development of several ontology modules aimed to address this general ontology design challenge. The resulting ontology is openly available [2], and additional documentation is available online [3]. As depicted in Figure 1,

---

[2] `https://github.com/ease-crc/soma`

[3] `https://ease-crc.github.io/soma/`



```
(perform
  (an action
    (type pouring-onto)
    (patient
      (some substance))
    (source
      (some container))
    (destination
      (some surface))))
```

Figure 1. SOMA represents physical and social context, and supports robotic agents in interpreting observed events, and realizing abstract descriptions.

our focus lies on representing both the physical context of realized everyday activities, as well as interpretations thereof as the social context.

2. Related Work

Ontology-based knowledge representation and reasoning in autonomous robot control is a fairly extensive field of research with developments in both service and industrial robotics. In the following, we will briefly discuss the most relevant works. A more detailed discussion about how ontologies are used to support robot autonomy is provided by Olivares et al. [5].

One example in the industrial robotics domain is the ROSETTA project [6,7]. Its initial scope was reconfiguration and adaptation of robot-based manufacturing cells, however, the authors have, since then, further developed their activity modeling for coping with a wider range of industrial tasks. Other authors have focused on modeling industrial task structure, part geometry features, or task teaching from examples [8,9, 10,11]. Compared to the everyday activity domain, industrial tasks considered in above works are more structured, and less demanding in terms of flexibility.

An approach to activity modeling in the service robotics domain is presented by Tenorth and Beetz [12]. Foundationally, their modeling is based on a subset of the discontinued OpenCyc ontology [13] with much weaker axiomatization compared to our foundational layer, and less inferential power and guidance during modeling. The scope of their work is similar to ours as the authors also consider how activity knowledge can be used to fill knowledge gaps in abstract instructions given to a robotic agent performing everyday activities. However, the scope of the work presented here is wider, as we also consider how activity knowledge can be used for the interpretation of observations. Our activity modeling is further more detailed in terms of activity structure as we also consider the processes and states that occur during an activity. Another difference is that, in their modeling, there is no distinction between physical and social context, but this dichotomy is central in SOMA.

A more general approach to activity modeling for robotic agents is presented by the IEEE-RAS working group ORA [14]. The group has the goal of defining a standard ontology for various sub-domains of robotics, including a model for object manipulation tasks. It has defined a core ORA ontology [15], as well as additional modules for industrial tasks such as kitting [16]. In terms of methodology, we differ in foundational assumptions we assert, which has important consequences on the structure of our ontology, modeling workflow, and inferential power. In the case of ORA, the SUMO upper-level ontology is used as foundational layer. However, the foundational layer of SUMO is rather weakly axiomatized compared to other models. In particular central in SOMA is the distinction between ground and descriptive concepts to represent physical and social activity context, and that this distinction is tightly coupled with the foundational layer.

3. Overview

In this section, we will discuss the scope of SOMA (Section 3.1), its underlying foundational commitments (Section 3.2), and how it is organized (Section 3.3).

3.1. Scope

The broad scope of our work is everyday object manipulation tasks in autonomous robot control, and in particular the motion and force characteristics of objects. The research question driving us is whether a single general control program can be written that can generate adequate behavior in many different contexts: for different tasks, objects, and environments. The employment of a general plan thus requires an abstract task and object model, and a mechanism to apply this abstract knowledge in situational context.

A more fine-grained scope is defined through a set of competency questions that are documented in the NEEM-Handbook [4]. Some examples related to the modeling of affordances are “what can an object be used for?” and “what can an object be used with?” (referring to the fact that affordances arise through the meeting of compatible dispositions), as well as “what cannot be used to manifest an affordance?”. Thus, the ontology offers ways to indicate what objects – given semantic knowledge about them – provably can or provably cannot be used for some purpose, with undecided cases being passed on to other mechanisms, e.g. simulation-based testing.

3.2. Foundational Commitments

SOMA is based on the DOLCE+DnS Ultralite (DUL) foundational framework [17]. This decision is greatly motivated by their underlying ontological commitments. Firstly, DUL is not a revisionary model, but seeks to express stands that shape human cognition. It assumes a multiplicative approach. Our work, however, seeks to apply a reductionist approach where possible – rather than capturing, for example, the flexibility of our usage of objects via multiple inheritance in a multiplicative manner, we commit to a reduced *ground* classification and use a *descriptive* approach for handling this flexibility, as provided by the addition of the *Descriptions and Situations* extension of DUL [18]. For this a primary branch of the ontology represents the ground **physical model**, e.g. objects and actions, while a secondary branch represents the **social model**, e.g. roles and tasks. All entities in the social branch are mind-dependent entities, i.e. they constitute social objects that represent concepts about, or descriptions of ground elements.

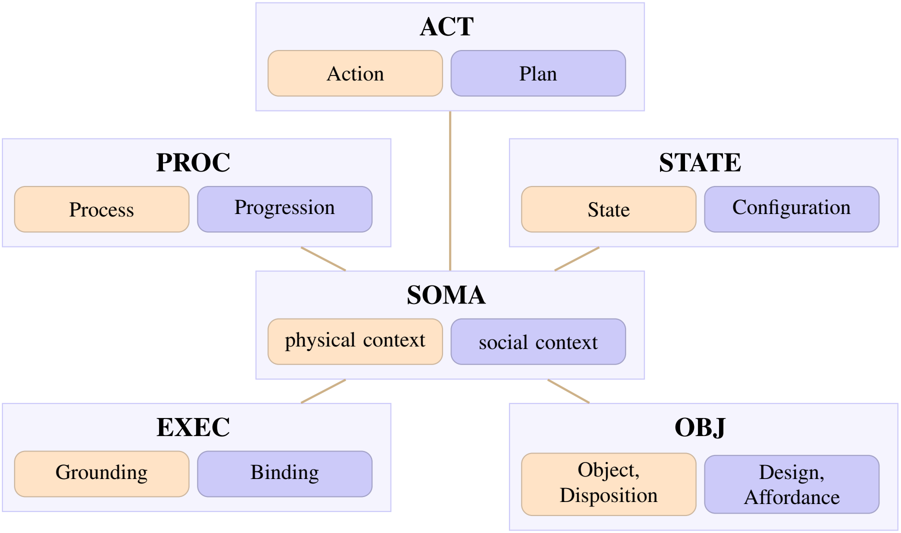


Figure 2. The modular organization of SOMA. Each module defines concepts and relationships used to represent physical (orange) and social (purple) activity context.

Every axiomatization in the physical branch can, therefore, be regarded as expressing some physical context whereas axiomatizations in the descriptive social branch are used to express social contexts. Already some dedicated relations and design patterns are provided that connect both branches[4]. For example, as detailed in Section 4.1, the DUL relation *classifies* connects ground objects, e.g. a hammer, with the roles they can play, while this does not represent classification in the logical sense, we find the distinction between what an object is and what it can be used for very suitable for our approach. Thus, we can state that a hammer can in some context be conceptualized as a murder weapon, a paper weight or a door stopper. Nevertheless, neither its ground ontological classification as a tool will change nor will hammers be subsumed as kinds of door stoppers, paper weights or weapons via multiple inheritance. Following a quick overview of the central modules of SOMA, we will provide detailed examples of where and how our commitments apply in Sections 4 and 5.

3.3. *Module Overview*

SOMA is organized in several modules that conceptualize different aspects of physical and social activity context (Figure 2). The different modules correspond to different event types (*ACT*, *PROC*, *STATE*), objects that participate in the activity (*OBJ*), and execution context (*EXEC*). The scope of each of these modules is outlined below.

The scope of the *OBJ* module is the representation of physical objects, and their qualities. The module includes two taxonomies used to classify objects: an object taxonomy in the grounded branch, and a role taxonomy in the descriptive branch. It further includes a taxonomy of dispositions to represent the potential of using an object in some way, and a design taxonomy used to categorize objects based on function, structure, and aesthetics. This will be described in more detail in Section 4.

---

[4] http://ontologydesignpatterns.org

The scope of the *ACT*, *PROC* and *STATE* modules is the contextualization of actions, processes and states. Actions are defined as events performed by an agent (physical context), and structured by a plan that is executed by the agent (social context). Plans may further impose constrains on steps in the plan, and objects that may play a role (*EXEC* module of SOMA). An action may cause processes to be started or stopped, and states to be changed. Processes, such as motions, are defined as events considered in their evolution. The difference between a state and a process is that, when considering time slices of the event, for states, these time slices always have the same type as the state (states are homeomeric), but for processes this is not the case. In SOMA, we define a taxonomy of event types in the descriptive branch used to classify actions, and to further decompose them into motion phases, state changes, and physical interactions caused by them. This will be described in more detail in Section 5.

4. Object Representation in SOMA

One of the reasons that everyday activity is a hard problem is the immense amount of variations an unrestricted environment may have, and the resulting potentials of interaction for an agent. Each type of object needs to be handled differently depending on its properties. However, object manipulation tasks are often defined independent of the type of object that is manipulated. It is thus crucial to employ an abstract object model, and a mechanism for applying abstract object knowledge to novel situations.

The main link between objects and actions in SOMA is that objects participate in events in the physical branch, and that the social branch represents the interpretation of their participation. This is elaborated in Section 4.1. Next, in Section 4.2, we will discuss how objects are organized along their design. However, the agent might further need to find suitable candidate objects to perform a task by reasoning about which objects have the potential to be used in a certain way. We employ an object disposition model for that purpose which is discussed in Section 4.3. An example object class is illustrated in Figure 3. In the example, a `Cup` is defined as a type of `DesignedContainer`, and thus inherits qualities through the class membership such as that it has a shape, and that it can be used as a container. However, a cup is specifically designed to contain liquid substances which can be captured by the notion of `FunctionalDesign` in SOMA. The full axiomatization of the concept may contain several similar statements to specify other aspects of object qualities such as that cups afford containment for other object classes too, that they have a specific structural design, etc. Such a definition can be exploited to formulate reasoning queries such as which known objects could be used for a particular purpose, e.g. for storing water, or what the potential uses of an object are. An example of such a reasoning query is provided in Section 6.1.

4.1. Object Types

For the classification of objects we employ the `Role` pattern provided by the foundational layer. Roles are `Concept`s and, as such, reside in the `SocialObject` branch of DUL. For human agents the ascription of roles to entities comes very natural. *He is a student* does not imply an *isa* or *instanceof* relation between some male individual and a student class. It is rather meant that at this point of his life the individual plays the role of a student, which, however, can and will change over time.

```
Class: LiquidContainmentDesign          Class: Cup
SubClassOf:                             SubClassOf:
  FunctionalDesign                        DesignedContainer
  isDesignOf only (hasDisposition         isDescribedBy some
    some (Containment                       LiquidContainmentDesign
    and (affordsTrigger some
        (classifies only Liquid)))
```

Figure 3. An example of how object classes are represented in SOMA.

This role pattern is of paramount importance, especially in the modeling of affordances discussed in Section 4.3. In the model presented herein, we import the roles that have been established in the field of frame semantics [19]. The selectional restrictions imposed by the *classifies* relation are used in a number of reasoning processes ranging from natural language understanding to tool selection. As certain roles can only classify physical agents or specific types of designed artifacts these axiomatizations provide substantial information about context dependent *meaning* of objects.

4.2. Object Designs

The organization of objects along a taxonomy is difficult as objects can be categorized in many ways. A notion of design is useful to capture object categories corresponding to structural, functional, or aesthetic patterns. Designs are in particular useful to conceptualize *refunctionalized* entities, and to support an agent to *hypothesize* unknown functions served by an entity. For example, a wooden pallet can be reused for the construction of furniture such as sofa, bed etc. The categorization of objects along their design can be employed in order to allow the use of more general plans, where, instead of object types, the plan refers to structure, aesthetics, or function.

Within the scope of SOMA, the `Design` concept belongs to the social branch. SOMA considers *structural*, *functional* and *aesthetic* aspects of design. A design describes classes of objects that host a common design-relevant quality. These qualities are dispositional, geometrical, and aesthetic aspects of the object. This corresponds to our design categorization into functional, structural, and aesthetic design. Each `Design` concept defines restrictions on the corresponding quality type that needs to be fulfilled by any object described by the design. These restrictions can also represent sufficient conditions under which an object is thought to be described by the design which allows the classification of entities given their design pattern can be detected.

In the scope of this work, we only consider functional aspects of objects. These are represented using a model of dispositional qualities which is discussed next.

4.3. Object Dispositions

Objects are important to an agent because they allow it to perform, or prevent it from performing, actions to achieve its goals. The notion of "affordance" was put forth by Gibson as "... what it offers the animal, what it provides or furnishes, either for good or ill" [20]. However, though evidently useful as a way to organize actionable knowledge about the world [21], affordances proved very difficult to model ontologically. Several approaches have been proposed, such as regarding affordances as qualities [22] or as events [23]. Nonetheless, we think these approaches are not satisfactory. Affordances are relational, characterizing a potential interaction of several objects, and therefore should not be treated as either a quality belonging to an object, nor as an event. We do recognize that some qualitative aspects of objects contribute to affordances, however, which is why we constructed our model around the interplay of Turvey's notion of *disposition* [24] and Gibson's notion of *affordance*.

In SOMA, the `Disposition` concept is defined as an object quality that allows an object to participate in events that *realize* an affordance. The `Affordance` concept itself, however, is defined as the relational context holding between several objects that play different roles such as being the "bearer", "trigger", or "background" of an affordance. Our modeling allows us, via a mixture of DL and other reasoning mechanisms such as simulation, to answer several interesting questions such as what affordances might an object provide in some combination with others, what objects might, or probably would not, be able to provide a given affordance, what combinations of objects would work towards providing an affordance etc. For more details on our disposition and affordance model, we invite the reader to consult our previous work on this topic [25].

5. Event Representation in SOMA

The information gap between an instruction given to an embodied agent and the way it has to move its body to *successfully* execute the instruction is often immense. Consider, for example, a recipe for cooking noodles that contains an instruction to *boil water in a pot*. It is simple to decompose this instruction into several steps with individual subgoals such as finding pot and tap, placing the pot underneath the tap, and filling the pot with water. However, the more difficult problem is how the agent has to move its body in each step such that the goal is achieved, and unwanted side-effects are avoided. Little variations in motion behavior may have drastic consequences in tasks that require delicate interaction. It is thus essential for agents performing actions in the physical world to reason about *how they should move* to achieve their goals in an appropriate, flexible and robust manner which is an unsolved problem for the general case.

SOMA attempts to support an agent facing this problem by equipping it with knowledge about relationships between abstract descriptions and their realization. The support is twofold. First, the agent may employ more general plans where informational gaps are filled by reasoning over knowledge represented with SOMA. Second, the agent may employ SOMA for understanding and generalizing observations. This means that agents can interact safer in environments with incomplete information, and that they can learn general patterns from specific situations.

An illustrative example of the representation of a pouring plan in OWL Manchester Syntax is provided in Figure 4. The plan is represented as an ABox ontology, i.e. as a collection of facts about the plan: what task it defines, and what steps it describes. Steps are conceptualizations of the events that realize them. They specify the roles objects need to play during the event, and may further specify ordering constraints using Allen's relations such as that realizations of the step *Pouring0* are *started by* realizations of the *Approaching0* step. Finally, resources may need to be shared among different steps within a plan, for example the source from which is poured (with role *Source0*) is the

```
Individual: PouringPlan_0               Individual: Approaching_0
  Types: Plan, Description                Types: Motion Type, Concept
  Facts: defines Pouring_0,               Facts: usesRole Destination_1,
         hasPhase Approaching_0,                 overlapsWith Tilting_0
         hasPhase Tilting_0             Individual: Tilting_0
Individual: Pouring_0                     Types: Motion Type, Concept
  Types: Task, Concept                    Facts: usesRole Patient_1
  Facts: usesRole Patient_0,            Individual: Binding_1
         usesRole Source_0,               Types: Role Binding,
         usesRole Destination_0,                 Description
         startedBy Approaching_0          Facts: hasBinding Source_0,
                                                 hasBinding Patient_1
```

Figure 4. An example of how plans are represented in SOMA.

same entity as the patient during the tilting motion (with role *Patient0*) which is captured through a role binding in the plan definition.

The reason that plans are represented as ABox ontologies in SOMA is that identity constraints cannot be expressed as OWL-DL axioms, i.e. distinct steps of a plan with the same type cannot be defined by different axioms. However, sequencing information can be encoded in the ABox. This has the drawback that an OWL reasoner cannot recognize the plan that was executed by an agent. In general, the machinery necessary to perform, or recognize the execution of a task is outside the scope of OWL-DL. Nonetheless, we have committed to encoding as many constraints on tasks as we can via OWL-DL axioms.

In the following, we will first introduce our hierarchical organization of tasks, processes and states in Section 5.1, and, second, how they are decomposed into phases with explicit goals and individual knowledge pre-conditions in Section 5.2. Finally, we will discuss our modeling of force dynamical characteristics in Section 5.3.

5.1. Event Types

One of the most important demands on a cognitive system is to reason about actions; colloquially speaking, an agent constantly asks itself what to do, and how to do it. This however opens up another question, namely what exactly is the entity that the agent represents – an actual event, or an interpretation of one.

As an example, consider this scenario: a robot moves toward a table carrying a plate. Midway, its gripper releases, dropping the plate, which shatters against the floor. Perhaps the robot had to transport the plate to the table, and it failed to do so; or perhaps it was required to drop the plate as part of some material test, and the table was just there for some other reason. Just by observation of the action, without other interpretive context which includes knowledge of what the robot was told to do, there is no reliable way to tell. The failed transport interpretation does seem more likely a priori, but only because we have more often seen people tell robots to transport plates rather than break them; we still make use of an expected interpretive context.

As a result, we do not define a taxonomy of action events in our ontology, but rather of tasks that are used to conceptualize actions. For example, the `Grasping` concept is defined as task in SOMA, and it is used for the *classification* of events that are interpreted as an intentional grasping activity. This classification pattern between events and their conceptualization is provided by the foundational layer of SOMA. However, within the foundational layer, this pattern is only instantiated for actions and their conceptualization. In our modeling, motions of an agent and other processes, as well as state events are used to structure an activity. Thus, we also need to represent processes and states in the ground and the descriptive ontology. The same pattern applies: the concepts `Process` and `State` are defined in the ground ontology, and their conceptualization in the descriptive ontology, and a relationship between both branches is established through the aforementioned classification pattern.

5.2. Event Phases

Actions in SOMA are composed of distinct *phases*. Each phase has its individual goal, and requires a different movement strategy to be executed successfully. The phases correspond to different stages of an object manipulation task, usually separated through *contact events*. Flanagan et al. have pointed out the importance of contact events in object manipulation tasks [26]. The authors have shown that contact events cause a distinct pattern in sensory events, and that they can be used as *sensorimotor control points* for aligning and comparing predictions with actual sensory events. Another justification is that humans have shown to direct their gaze to contact points when they perform object manipulation tasks, or when they observe another agent performing a task.

The structure of activities in SOMA is governed by a set of design patterns. At its core, SOMA activity modeling builds on top of the *basic plan* ontology design pattern that represents plans and their execution. The pattern defines that an execution is a situation that *satisfies* the description of the plan. However, the pattern is defined too specific for the scope of this work, as we also want to provide descriptive context for states and processes. Hence, we generalize this pattern such that it can be instantiated for actions, states and processes: An `Action` is described in a `Plan` which is a description having an explicit goal. A plan satisfies situations that include action sequences that match the structure of the plan, such situations are called `Plan Executions`; a `State` is described in a `Configuration` which includes constraints on regions of entities and relationships between them. A configuration satisfies situations in which all constraints of the configuration are satisfied; and a `Process` is described in a `Process Flow` which is a description of the progression of the process. A process flow satisfies situations that include a process that progresses in the described way. Another aspect of activity structure can be captured by SOMA in, what we call, *execution contexts*. These are representations of how different phases of an activity constrain each other depending on conditions encountered in the activity execution. In particular, we define the `Binding` concept as identity constrain representing that a parameter or role grounding is the same in different phases, however potentially being classified differently.

Ordering constraints are expressible in SOMA through a sequence pattern based on Allen's interval calculus [27]. Allen's calculus defines thirteen relations between time intervals including *before*, *after*, *overlaps*, and *meets*. This is useful to, on the one hand, represent precedence of one phase strictly following the other, and, on the other hand, it allows to cope with concurrency in the sequence. We apply this algebra to event types that are defined within the descriptive context of a plan or process flow. However, reasoning about sequences is not well supported in OWL. Instead, we translate interval relations into a point graph to perform point-based reasoning [28]. Point graphs are directed acyclic graphs where nodes are the endpoints of intervals, and an edge is added for each axiom $a < b$ where a,b are interval endpoints. A non empty path from an endpoint a to another endpoint b further implies that $a < b$ through the transitivity of the relation. Event relations can be inferred through relations between their endpoints. For example, an interval i_1 *precedes* another interval i_2 iff $e_1 < s_2$ where s_2 is the starting point of i_2 and e_1 is the ending point of i_1. However, this only covers the *pointisable* subclass of the algebra which means that e.g. disjunction axioms are not expressible.

Knowledge about the structure of activities can be employed by an agent in both directions: for planning an activity, and for interpreting observed events. Planning can be seen as a mapping from the descriptive to the grounded branch of SOMA, while interpretation maps the other way. For embodied agents, planning goes beyond mere decomposition of an activity into steps, the agent may further need to decide what objects it should use, how it should move, with what speed, and how much force it should apply when getting into contact with some object. SOMA may be employed by the agent to find potential sequences of steps and motions to execute a task, to support finding potential objects playing some role during the activity, and to constrain the values of parameters of a task. Interpretation of events, on the other hand, is often possible through detection of contact events, types of motions, and states. These can be used as tokens for an activity parser that uses SOMA as a grammar, this will be described more in Section 6.

Knowledge Pre-Conditions. In order to execute a motion, an agent has to invoke one of its control routines with a set of arguments. Higher-level routines may have a notion of object, but at a lower-level all boils down to numbers such as with what effort the robot moves, how fast, etc. SOMA allows to define constraints for both cases: for the types of objects that can play a role during the action, and for the value of parameters. This is done by using restrictions on what types of objects or regions can be classified by some role or parameter. This information is used to reduce the search space for doing an appropriate object or parameter selection (Section 4.3).

Goals. A goal is a description of a desired situation, and it is achieved only if the situational context, after the execution has been finished, satisfies this description. SOMA is more specific about what it means to execute an action successfully as it decomposes it into processes and states where the goal of the task is that the progression of processes evolves, and that state changes occur as described. Particularly important are the contact states in object manipulation tasks, as they represent control points for the agent when generating or observing behavior.

5.3. Event Force Characteristics

A contact state is an indicator for whether objects are touching each other or not. Patterns of such states are useful for distinguishing between categories of activities. However, different activities may cause the same pattern while their goal is different, or even the opposite of each other. This is, for example, the case for pulling and holding. Both cause the same pattern of an endeffector getting into contact with another object. But the force characteristics are different: the goal of a pulling task is to overcome the inertial force of the object to set it into motion, and the goal of a holding task is to neutralize any external force that would set the object into motion. Another aspect is that an agent performing such a task needs to decide how much force to apply. In order to make this decision it is valuable to know what the intended force-related consequences are.

SOMA supports the representation of force characteristics using Talmy's notion of force dynamics [29]. Talmy distinguishes between two entities that participate in force dynamical processes: the `Agonist`, and the `Antagonist`. An agonist is the subject of a force dynamical expression, while the antagonist is the opposing force in the expression. Each expression has an intrinsic force tendency either to set the agonist into motion, or to keep it resting. Whether the tendency can be realized or not depends on which of the two entities is the *stronger* entity.

6. Evaluation of SOMA

SOMA was developed to provide robots with the capability to answer a set of competency questions about everyday activities. Thus, we evaluate SOMA by validating that these questions can be answered (in Section 6.1). Due to space limits, we will only elaborate on selected examples here. The full range of competency questions is documented in the NEEM-Handbook [4]. Furthermore, the relevance of these competency questions can be demonstrated through applications of SOMA and their evaluation. A practical employment of SOMA is demonstrated in the *EASE Robot Household marathon* [30]. Here, we will provide an overview of SOMA applications in prior work to verify its use in the application domain of autonomous robotics (in Section 6.2).

In version 1.1.0, SOMA contains 1330 logical axioms, 416 classes, 203 object properties, and 38 data properties. Its expressivity is SROIQ(D). More metrics are listed on the SOMA webpage [5]. They are automatically computed when SOMA is deployed through a web service based on OntoMetrics [6].

6.1. Reasoning with SOMA

Being written in DL, SOMA can be processed with standard DL reasoners such as HermiT. Because reasoning with the ontology is important during its use, our process of updating the ontology includes a reasoning step as well, also performed with HermiT, to verify that updates do not insert unsatisfiable concepts or empty properties. In more detail, every commit to the SOMA repository triggers subsumption and classification queries, and the discovery of concepts or properties equivalent to `Nothing` triggers a warning. This eases maintenance and scaling up of SOMA while keeping it consistent.

We will next exemplify how the ontology can be reasoned with "at runtime", during some activity of a robot. Knowledge in SOMA covers, among others, aspects such as dispositions and affordances of objects. A question a robot might have is, what object in its environment could be used for a particular purpose, e.g. to contain some liquid. To this end, the robot will query the ontology by first defining a new "query" concept, formulated in Listing 1 and then ask which of the objects it knows about can be proven to belong to this query concept via a subsumption query – objects that are individuals of subconcepts of the query concept can be used.

---

[5] `https://ease-crc.github.io/soma/`

[6] `https://ontometrics.informatik.uni-rostock.de`

Sometimes, no known objects might be provably appropriate for a purpose. In such cases, one might try some other methods, such as testing in simulation, but such methods are themselves costly and so a filtering of candidates via reasoning is useful. In this example, the robot might ask, "what cannot be used to contain a liquid". This is also achieved with the help of the "query" concept illustrated in Listing 1, but in a different manner. For a named concept `C` present in SOMA, that is a subconcept of `DesignedArtifact`, do a satisfiability query for the intersection of `C` with the query concept. If this intersection is provably empty, objects that are instances of `C` need not be tested for the affordance.

Listing 1: A query concept to find objects which can contain liquids

```
Class: WithAffordance_Containment_Liquid
EquivalentTo:
  DesignedArtifact and (hasDisposition some
    (Containment and (affordsTrigger some (classifies only Liquid))))
```

OWL-DL was designed for the representation of encyclopedic knowledge, and has limited scope for domains such as dynamical characteristics. This concerns, for example, reasoning about the temporal ordering of steps that execute a task which we handle through *point-based* reasoning where SOMA is enriched with definitions of temporal relations. Another example is *simulation-based* reasoning for affordance testing [31,32]. Thus, SOMA is used as a common model in a hybrid reasoning framework.

6.2. Applications of SOMA

The applicability of SOMA in the domain of autonomous robotics has been demonstrated in several scientific publications which we will briefly discuss in this section.

Grounding task parameters often requires predictive models which can be trained over instances of successful performance. Such experiential knowledge is in particular useful to learn context-dependent *plan specializations*. That is, how the parameters of the plan can be constrained within the scope of some context to reduce the search space of parameter selection during plan execution. The learning problem is then defined with respect to a contextual pattern, and experiential samples are only considered when their contextualization matches this pattern. We have demonstrated this capability in another work where a robot learns to execute a general fetch and place plan based on experience acquired through the execution of more constrained tasks [33].

Learning mechanisms often require large amounts of training data. One modality for acquisition is observation of other agents. We have developed an activity parser that is used to find possible interpretations for observed patterns of occurrences such as that objects get into contact with each other, or that the state of an object changed. The grammar used by the parser is a library of plans represented using SOMA. In prior work, we have provided more details about how the social context in SOMA can be grounded in data structures of a game engine [34]. The game engine implements an immersive virtual reality environment with photo realistic rendering and state of the art physics engine. Users perform object manipulation tasks while interactions, states, and motions are monitored, and used as tokens by the activity parser.

Our modeling of tasks also helps to disambiguate vague natural language commands a robot may receive. SOMA allows us to model how tasks relate to and depend on one another, and thus define execution contexts containing not just information about a task's parametrization, but also information about what other tasks it should enable. We use such execution contexts to set up simulation scenarios in which to test task executions and thus select among several interpretations of a vague natural language command [31].

7. Conclusion

In this paper, we have proposed SOMA, a novel activity ontology for robotic agents that combines several established ontology design patterns with models of human cognition. The SOMA ontology has been designed to cover a set of competency questions in order to support robot decision making during action execution or observation. This is done by representing physical and social context of an activity, and by establishing relationships between both contextualizations. These representations are used by robotic agents to fill knowledge gaps in general plans applicable to many situations, and to generate context-specific behavior. They are further used for the representation of observed events, and for reasoning about how they are to be interpreted. We believe that such an expressive activity representation is an important vehicle for transforming robots from just performing a task to mastering the corresponding activity.

References

[1] Lenat DB, Feigenbaum EA. On the thresholds of knowledge. Artificial Intelligence. 1991;47(1):185 250. Available from: `http://www.sciencedirect.com/science/article/pii/000437029190055O`.

[2] Anderson JE. Constraint-Directed Improvisation for Everyday Activities. The University of Manitoba (Canada); 1995. AAINN99082.

[3] Beetz M, Beßler D, Haidu A, Pomarlan M, Bozcuoglu AK, Bartels G. KnowRob 2.0 – A 2nd Generation Knowledge Processing Framework for Cognition-enabled Robotic Agents. In: International Conference on Robotics and Automation (ICRA). Brisbane, Australia; 2018. p. 512-9.

[4] Beetz M, Beßler D, Koralewski S, Mihai P, Vyas A, Hawkin A, et al.. NEEM Handbook; 2020. Accessed: 2021-04-28. `https://ease-crc.github.io/soma/owl/1.1.0/NEEM-Handbook.pdf`. Available from: `https://ease-crc.github.io/soma/owl/current/NEEM-Handbook.pdf`.

[5] Olivares-Alarcos A, Beßler D, Khamis A, Goncalves P, Habib M, Bermejo J, et al. A Review and Comparison of Ontology-based Approaches to Robot Autonomy. The Knowledge Engineering Review. 2019;34:e29.

[6] Patel R, Hedelind M, Lozan-Villegas P. Enabling robots in small-part assembly lines: The ”ROSETTA approach” - an industrial perspective. In: ROBOTIK. VDE-Verlag; 2012. .

[7] Stenmark M, Malec J, Nilsson K, Robertsson A. On Distributed Knowledge Bases for Robotized Small-Batch Assembly. IEEE Transactions on Automation Science and Engineering. 2015;12(2):519-28.

[8] Balakirsky S. Ontology based action planning and verification for agile manufacturing. Robotics and Computer-Integrated Manufacturing. 2015;33(Supplement C):21 28. Special Issue on Knowledge Driven Robotics and Manufacturing.

[9] Polydoros AS, Großmann B, Rovida F, Nalpantidis L, Krüger V. Accurate and Versatile Automation of Industrial Kitting Operations with SkiROS. In: Towards Autonomous Robotic Systems - 17th Annual Conference (TAROS); 2016. p. 255-68.

[10] Kootbally Z, Schlenoff C, Lawler C, Kramer T, Gupta SK. Towards Robust Assembly with Knowledge Representation for the Planning Domain Definition Language (PDDL). Robot Comput-Integr Manuf. 2015 Jun;33(C):42-55.

[11] Perzylo A, Somani N, Profanter S, Kessler I, Rickert M, Knoll A. Intuitive instruction of industrial robots: Semantic process descriptions for small lot production. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2016. p. 2293-300.

[12] Tenorth M, Beetz M. Representations for robot knowledge in the KnowRob framework. Artificial Intelligence. 2015.

[13] Lenat D. CYC: A Large-Scale Investment in Knowledge Infrastructure. Communications of the ACM. 1995 November;38(11):33-8.

[14] Schlenoff C, Prestes E, Madhavan R, Goncalves P, Li H, Balakirsky S, et al. An IEEE standard ontology for robotics and automation. In: IEEE Int. Conf. on Intelligent Robots and Systems (IROS); 2012. p. 1337-42.

[15] Prestes E, Carbonera JL, Fiorini SR, Jorge VAM, Abel M, Madhavan R, et al. Towards a core ontology for robotics and automation. Robotics and Autonomous Systems. 2013;61(11):1193 1204. Ubiquitous Robotics.

[16] Fiorini SR, Carbonera JL, Gonçalves P, Jorge VAM, Rey VF, Haidegger T, et al. Extensions to the Core Ontology for Robotics and Automation. Robot Comput-Integr Manuf. 2015 Jun;33(C):3-11.

[17] Masolo C, Borgo S, Gangemi A, Guarino N, Oltramari A. WonderWeb Deliverable D18 Ontology Library (final). IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web; 2003.

[18] Gangemi A, Mika P. Understanding the Semantic Web through Descriptions and Situations. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE - OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003; 2003. p. 689-706. Available from: `https://doi.org/10.1007/978-3-540-39964-3_44`.

[19] Baker CF, Fillmore CJ, Lowe JB. The Berkeley FrameNet Project. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1. ACL '98/COLING '98. USA: Association for Computational Linguistics; 1998. p. 86–90. Available from: `https://doi.org/10.3115/980845.980860`.

[20] Gibson JJ. The Ecological Approach to Visual Perception. Psychology Press Classic Editions; 1979.

[21] Yamanobe N, Wan W, Ramirez-Alpizar IG, Petit D, Tsuji T, Akizuki S, et al. A Brief Review of Affordance in Robotic Manipulation Research. Journal of the Robotics Society of Japan. 2018 01;36:327-37.

[22] Ortmann J, Kuhn W. Affordances As Qualities. In: Proceedings of the 2010 Conference on Formal Ontology in Information Systems: Proceedings of the Sixth International Conference (FOIS 2010). Amsterdam, The Netherlands, The Netherlands: IOS Press; 2010. p. 117-30.

[23] Moralez LA. Affordance ontology: towards a unified description of affordances as events. Res Cogitans. 2016;7(1):35-45.

[24] Turvey MT. Ecological foundations of cognition: Invariants of perception and action. American Psychological Association. 1992.

[25] Beßler D, Porzel R, Pomarlan M, Beetz M, Malaka R, Bateman JA. A Formal Model of Affordances for Flexible Robotic Task Execution. In: ECAI. vol. 325 of Frontiers in Artificial Intelligence and Applications. IOS Press; 2020. p. 2425-32.

[26] Flanagan JR, Bowman MC, Johansson RS. Control strategies in object manipulation tasks. Current Opinion in Neurobiology. 2006.

[27] Allen JF. Maintaining knowledge about temporal intervals. In: In Communications of the ACM, Volume 26 Issue 11; 1983. .

[28] Zaidi AK, Wagenhals LW. Planning temporal events using point–interval logic. Mathematical and Computer Modelling. 2006;43(9):1229-53. Optimization and Control for Military Applications.

[29] Talmy L. Toward a Cognitive Semantics. Volume 2: Typology and Process in Concept Structuring. Language, Speech, and Communication. Cambridge, MA: MIT Press; 2000.

[30] Kazhoyan G, Stelter S, Kenfack FK, Koralewski S, Beetz M. The Robot Household Marathon Experiment. In: IEEE International Conference on Robotics and Automation (ICRA). Xi'an, China; 2021. Available from: `https://arxiv.org/abs/2011.09792`.

[31] Bateman J, Pomarlan M, Kazhoyan G. Embodied Contextualization: Towards a Multistratal Ontological Treatment. Applied Ontology. 2019;14(4):379-413.

[32] Pomarlan M, Bateman J. Embodied Functional Relations: a Formal Account Combining Abstract Logical Theory with Grounding in Simulation. In: Proceedings of the 11th International Conference on

Formal Ontology in Information Systems. FOIS; 2020. .

[33] Koralewski S, Kazhoyan G, Beetz M. Self-specialization of General Robot Plans Based on Experience. Robotics and Automation Letters. 2019.

[34] Haidu A, Beßler D, Bozcuoglu AK, Beetz M. KNOWROB-SIM — Game Engine-enabled Knowledge Processing for Cognition-enabled Robot Control. In: International Conference on Intelligent Robots and Systems (IROS). Madrid, Spain: IEEE; 2018. p. 4491-8.

Formal Ontology in Information Systems
F. Neuhaus and B. Brodaric (Eds.)

Author Index

This page intentionally left blank