

Upper Ontologies in COLORE

Michael GRÜNINGER^{a,1}, Carmen CHUI^a and Megan KATSUMI^a

^a*Department of Mechanical & Industrial Engineering, University of Toronto*

Abstract. This paper explores an alternative vision for upper ontologies which is more effective at facilitating the sharability and reusability of ontologies. The notion of generic ontologies is characterized through the formalization of ontological commitments and choices. Ontology repositories are used to modularize ontologies so that any particular upper ontology is equivalent to the union of a set of generic ontologies. In this way, upper ontologies are not replaced but rather integrated with other theories in the ontology repository.

Keywords. upper ontologies, ontology design, ontology verification, modularization, ontology repository, COLORE, DOLCE, SUMO, TUpperWare

1. Introduction

Within the applied ontology community, upper ontologies are widely recognized as tools to support the tasks of ontology design and semantic integration. In both of these tasks, the existence of unintended models (through an axiomatization that is too weak) or the omission of models (through an axiomatization that is too strong) can lead to serious problems. With semantic integration, for example, we must be able to guarantee that the inferences made with sentences exchanged between software applications are equivalent to the inferences made with respect to the applications' intended models – given some input the application uses these intended models to infer the correct output. The existence of unintended models can therefore prevent semantic integration.

Verifying that upper ontologies do not have unintended models (which can be ruled out with further axiomatization), and that they are not missing intended models, is therefore critical to the application of upper ontologies. This paper summarizes results about the application of the mathematical ontologies within the Common Logic Ontology Repository (COLORE)² [7] for the verification of upper ontologies, as well as the application of the generic ontologies within COLORE for the specification of mappings between upper ontologies and the design of new upper ontologies. After providing background material on the use of an ontology repository for the design and verification of upper ontologies, we review the extensive work that has been done in the verification and modularization of two upper ontologies, the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE)³ and Suggested Upper Merged Ontology (SUMO)⁴. We then show how the translation definitions used for the verification of the upper on-

¹Corresponding Author: Michael Grüninger; E-mail: gruninger@mie.utoronto.ca

²<http://colore.oor.net/>

³<http://www.loa.istc.cnr.it/old/DOLCE.html>

⁴<http://www.adampease.org/OP/>

ologies allows for the automatic generation of direct semantic mappings between the ontologies themselves. We finish the paper with the design and verification of a new upper ontology based on the generic ontologies within COLORE, which will support the semantic integration of international standards.

2. Building Upper Ontologies within a Repository

Upper ontologies have traditionally arisen from the approach in which concepts that are common across a set of domains can be axiomatized at a general level. The rationale is that reuse across domains is to be supported through specialization of the general concepts from an upper ontology. Similarly, semantic integration between ontologies is to be achieved through the general concepts they specialize. The work of [8] presented an alternative perspective (referred to as the *sideways approach*) to the conventional upper ontology paradigm. Rather than think of an upper ontology as a monolithic axiomatization centred on a taxonomy, the sideways approach considers an upper ontology to be a modular ontology composed of generic ontologies that cover concepts including those related to time, process, and space. The verification of the upper ontology follows from the verification of these generic ontologies, and the metalogical property known as *reducibility*.

In this section, we will review the basic principles of COLORE that are needed in order to characterize upper ontologies and their verification. The key techniques are based on the identification of the different relationships between ontologies – *nonconservative extension* for ontologies with the same signature, and *synonymy* for ontologies that have different signatures. We will later see how this supports the specification of mappings between upper ontologies.

2.1. Generic Ontologies

The basic organizational principle in COLORE is the notion of a hierarchy [7], which is a set of ontologies⁵ with the same signature.

Within COLORE, there are many ontologies which axiomatize classes of mathematical structures such as orderings, graphs, groups, and geometries. In the hierarchies of such mathematical ontologies, the root theory axiomatizes an elementary class of structures, and in principle, we can consider any consistent extension of the root theory to axiomatize some subclass of the structures. On the other hand, COLORE also contains many ontologies that axiomatize generic domains such as time, process, space, and shape. How can we distinguish between generic ontologies (which are intuitively about concepts) and the mathematical ontologies?

The work of [8] used the relationships among ontologies to propose necessary conditions for generic ontologies through a formalization of the notions of ontological commitments and ontological choices. While the ontological commitments are formalized by the set of axioms that are inherent in a particular generic concept, i.e. the assumptions a

⁵We follow previous work in terminology and notation treating ontologies as logical theories [7]. We do not distinguish between logically equivalent theories. For every theory T , $\Sigma(T)$ denotes its signature, which includes all the constant, function, and relation symbols used in T , and $\mathcal{L}(T)$ denotes the language of T , which is the set of first-order formulae that only use the symbols in $\Sigma(T)$.

relation must satisfy to be referred to by the concept’s name, ontological choices capture the idea that certain additional assumptions may be desirable in a particular domain or application, but which are not satisfied for all the various interpretations of the particular generic concept. In terms of COLORE, ontological commitments are axiomatized within the root theory of the hierarchy and the ontological choices are axiomatized by the trunk theories of the hierarchy.

2.2. Redefining Upper Ontologies

The central claim of [8] is that an upper ontology is a reducible ontology that has a reduction whose reductive modules are all generic ontologies. Each upper ontology is therefore composed of a set of generic ontologies, and each generic ontology axiomatizes a particular set of generic concepts (e.g., the classes and relations relevant for time, process, and space).

This approach leads to two observations. First, the verification of an upper ontology leads to an understanding of the conceptual scope of the upper ontology as the set of generic ontologies that are its reductive modules. This in turn allows us to compare upper ontologies both in terms of coverage but also in terms of the axiomatizations of their generic ontology modules.

Whereas the above discussion focuses on the *analysis* of upper ontologies, the second observation that we can make is about the *design* of upper ontologies. New upper ontologies can be designed by the union of different generic ontologies that already exist within the ontology repository. We will apply this idea in the last section of this paper to design and evaluate a new upper ontology.

3. Verification of Upper Ontologies

In this section, we review the work that has been done in ontology verification using COLORE, as applied to upper ontologies (DOLCE and SUMO) and related generic ontologies (e.g., OWL-Time). Through verification, we characterize the models of an ontology up to isomorphism. For upper ontologies, we do not do this directly, but rather we specify mappings between the upper ontology and existing mathematical ontologies in COLORE whose models have already been characterized up to isomorphism.

Definition 1 *Let T_0 be a theory with signature $\Sigma(T_0)$ and let T_1 be a theory with signature $\Sigma(T_1)$ such that $\Sigma(T_0) \cap \Sigma(T_1) = \emptyset$. Translation definitions for T_0 into T_1 are conservative definitions in $\Sigma(T_0) \cup \Sigma(T_1)$ of the form*

$$\forall \bar{x} p_i(\bar{x}) \equiv \Phi(\bar{x})$$

where $p_i(\bar{x})$ is a relation symbol in $\Sigma(T_0)$ and $\Phi(\bar{x})$ is a formula in $\Sigma(T_1)$.

The key to this endeavour is the property of *logical synonymy*: two theories T_1 and T_2 are synonymous iff there exist two sets of translation definitions Δ and Π , respectively from T_1 to T_2 and from T_2 to T_1 , such that $T_1 \cup \Pi$ is logically equivalent to $T_2 \cup \Delta$. By the results in [13], there is a bijection on the sets of models for synonymous theories. We can

therefore characterize the models of the ontology being verified by demonstrating that the ontology is synonymous with a logical theory whose models we understand.

Ontology verification allows us to identify unintended models (and hence propose missing axioms) for the upper ontology, as well as identify omitted models (and hence recommend the removal of axioms) for the upper ontology. Of course, this immediately leads to the question of how we can determine that certain models of an upper ontology are unintended. Whereas domain ontologies (e.g., in bioinformatics and commerce) are often designed with respect to a set of semantic requirements and competency questions, upper ontologies are implicitly designed to support interoperability. Since generic ontologies form reductive modules of an upper ontology, the problem of identifying the unintended and omitted models of an upper ontology is equivalent to determining that certain models of a generic ontology are unintended. In the following, we will see how these ideas play out in the context of the generic ontologies for time and mereotopology within the SUMO and DOLCE upper ontologies.

3.1. SUMO

SUMO [14] is an open source formal foundational ontology axiomatizing, among others, general concepts such as those needed to represent temporal and spatial location, mereotopology, units of measure, objects and processes. As an upper ontology, it is expected to be used as a global reference for semantics, and its axiomatization reused in the construction of domain and application ontologies for automated reasoning in expressive languages. In addition to the main ontology, which contains about 4000 axioms, SUMO has been extended with a mid-level ontology and a number of domain specific ontologies.

3.1.1. SUMO-Time

The work of [11] focused on the core axioms of the TEMPORAL subtheory covering the complete axiomatization of time through intervals and points, which was referred to as *SUMO Time*. In this subtheory, both timepoints and time intervals are elements of the domain, which is similar to the ontological commitments made by generic time ontologies such as OWL-Time and Endpoints.

The central result of [11] is the verification of a nonconservative extension of SUMO-Time; that is, SUMO-Time was extended to a new ontology $T_{sumo_ordered_timepoints} \cup T_{sumo_timeintervals}$ that is logically synonymous with the mathematical ontologies:

$$T_{bounded_linear_ordering}^6 \cup T_{strict_graphical}^7$$

Without the additional axioms, this aforementioned Theorem cannot be proven, and this is equivalent to saying that there are unintended models of SUMO-Time. In this way, the verification of SUMO-Time in [11] led to the identification of three classes of unintended models – models with a unique timepoint, models with partially ordered sets of timepoints (rather than linearly ordered sets), and models in which time intervals are temporal parts of timepoints. Three new axioms were proposed to eliminate these unintended models, and then the extended theory $T_{sumo_ordered_timepoints} \cup T_{sumo_timeintervals}$ was verified, characterizing its models up to elementary equivalence.

⁶http://colore.oor.net/orderings/bounded_linear_ordering.clif

⁷http://colore.oor.net/bipartite_incidence/strict_graphical.clif

3.1.2. SUMO-Mereotopology

SUMO also contains a subtheory that axiomatizes a *part* relation, together with other mereological relationships such as the sum, product, and difference of elements. The work of [12] and [15] provide a characterization of the models of this subtheory. As was the case with SUMO-Time, the verification led to the identification of unintended models, so that a nonconservative extension of SUMO was proposed. It also led to the identification of omitted models, that is, the axiomatization of SUMO eliminated intended models, namely, mereologies that are not equivalent to linear orderings. A proposal was made to weaken the axioms, and the resulting subtheory was shown to be synonymous the mereology $T_{comp_mereology}$.

3.2. DOLCE

The Descriptive Ontology for Linguistic and Cognitive Engineering DOLCE [6] [10] originated as part of the WonderWeb project⁸, which aimed to provide the infrastructure required for a large-scale deployment of ontologies intended to be the foundation for the Semantic Web. The work of [9] was the first to demonstrate the consistency of the DOLCE upper ontology, but it only proved the existence of a model rather than provide a characterization of *all* of the models of DOLCE. [2] and [3] provide the verification of DOLCE. Unlike the case with SUMO, no unintended models were discovered, and hence no additional axioms were proposed. A wide range of mathematical ontologies are required to characterize the models of DOLCE.

4. TUpperWare

As we observed earlier in the paper, new upper ontologies can be designed by the union of different generic ontologies that already exist within the ontology repository. In this section, we propose a verified ontology whose reductive modules are generic ontologies in COLORE.

4.1. Generic Ontology Modules

The goal for TUpperWare is to support the ontological analysis of relevant existing standards and to integrate the ontologies within those standards, in particular:

- ISO 18629 (PSL)⁹
- OWL-Time¹⁰
- ISO 19150 (GeoSPARQL)¹¹, ISO 19107¹²
- Semantic Sensor Network (SSN) Ontology¹³

In this sense, TUpperWare will be a requirements-driven upper ontology – the axiomatization must be strong enough to interpret the intended semantics of the terminology of each of the participating standards.

⁸<http://wonderweb.semanticweb.org>

⁹<https://www.iso.org/standard/35431.html>

¹⁰<https://www.w3.org/TR/owl-time/>

¹¹<http://www.opengeospatial.org/standards/geosparql>

¹²<https://www.iso.org/standard/26012.html>

¹³<http://purl.oclc.org/NET/ssnx/ssn>

Ontological Category	Generic Ontology Module
time	http://colore.oor.net/combined_time/interval_with_endpoints.clif
activity, participation, state	http://colore.oor.net/process_specification_language/psl_outcore.clif
duration	http://colore.oor.net/duration/point_duration.clif
space, boundaries	http://colore.oor.net/multidim_mereotopology_codib/codib.clif
location, physical bodies	http://colore.oor.net/occupy/occupy_root.clif
shape	http://colore.oor.net/boxworld/boxworld.clif
mereotopology of physical objects	http://colore.oor.net/component/component.clif
physical objects	http://colore.oor.net/opo/opo.clif

Table 1. Ontological categories and modules within the TUpperWare ontology.

To this end, TUpperWare currently consists of the generic ontology modules shown in Table 1. The scope of TUpperWare is determined by the task of *standards integration* – each of the generic ontologies is needed to support one of the participating standards. Each of the generic ontologies is a reductive module of TUpperWare. There are no additional bridge axioms in a combined signature across these modules. The verification of TUpperWare is therefore trivial, being equivalent to the verification of each of the generic ontologies.

TUpperWare also contains modules for reasoning about state designed using the methodology of [1], so that domain state ontologies and domain process ontologies are specified based on corresponding generic ontologies. For example, the location module T_{occupy} within TUpperWare leads to the motion ontology (i.e. how location changes), the mereotopology of physical objects leads to the assembly ontology (i.e. how components of a physical object change), and the ontology of physical objects T_{opo} leads to physical process ontology. This latter ontology includes the axiomatization of temporary parthood.

5. Modularization of Upper Ontologies

The reduction of a theory can also be used to decompose an ontology [7]. If T is reducible to S_1, \dots, S_n , then there exist subtheories T_1, \dots, T_n such that each T_i is synonymous with S_i . Since T is a conservative extension of each T_i , we refer to the subtheories as the reductive modules of T . We can therefore use the results from verification to modularize upper ontologies.

The verification of SUMO-Time led to its decomposition into two modules – $T_{sumo_ordered_timepoints}$ axiomatizes the linear ordering over timepoints, and $T_{sumo_timeintervals}$ axiomatizes the relationship between timepoints and time intervals. An interesting outcome was a restructuring of SUMO-Time as the definitional extension of two modules, thus greatly simplifying the axiomatization.

The verification of DOLCE led to a modularization that was strikingly different from the modularization that was used in the consistency proof by [9]. Rather than sets of axioms for relations such as constitution and temporary parthood, the reductive mod-

ules of DOLCE are subtheories for classes of elements – perdurants, physical endurants, nonphysical endurants, and qualities.

On the other hand, the TUpperWare Ontology itself is explicitly designed in a modular fashion (i.e. by combining subtheories together); there is no need to modularize TUpperWare a posteriori.

TUpperWare Module	DOLCE Module
activity	Perdurant ($T_{dolce_perdurant}$)
participation, state	Participation ($T_{dolce_participation}$)
state	–
duration	–
space, boundaries	?
location, physical bodies	?
shape	?
mereotopology of physical objects	Mereology ($T_{dolce_mereology}$)
physical objects	Physical Endurant (T_{dolce_PED})
motion ontology	–
assembly process ontology	–
physical process ontology	–
?	Nonphysical Endurant (T_{dolce_NPED})
?	Dependence ($T_{dependence}$)
–	Quality ($T_{dolce_Quality}$)

Table 2. Comparison of modules in the TUpperWare and DOLCE ontologies. A dash (–) indicates that no there is no corresponding module. A question mark (?) indicates an open problem as to whether a corresponding module exists.

Table 2 is a comparison of the modules of TUpperWare and the reductive modules of DOLCE. The primary differences between the modularization of the two ontologies arise from the way they treat space, shape, and location. Within DOLCE, these are treated as qualities, but it currently appears that this is a fundamentally different from the representation in TUpperWare.

6. Mappings between Upper Ontologies

The area of ontology mappings covers a diverse range of techniques [5]. What do we mean by mappings between ontologies? First, we are specifying mappings between modules of the upper ontologies. Second, we are identifying synonymous subtheories of each pair of modules. Note that we are using the axioms alone – we are not using any prior intuitive understanding of the concepts involved.

Mappings between upper ontologies are automatically generated from the translation definitions used in the verification of the upper ontologies [4]. Since each generic ontology is synonymous with a mathematical theory, we can exploit the transitivity of synonymy to compose mappings and identify the synonymous “images” within each hierarchy.

Suppose T_1, T_2 are two different generic ontologies, and S_1, S_2 are two mathematical ontologies in the same hierarchy such that T_1 is synonymous with S_1 and T_2 is

synonymous with S_2 . By the definition of synonymy, there exist translation definitions $\Delta_1, \Pi_1, \Delta_2, \Pi_2$ such that

$$\begin{aligned} T_1 \cup \Delta_1 &\models S_1 & S_1 \cup \Pi_1 &\models T_1 \\ T_2 \cup \Delta_2 &\models S_2 & S_2 \cup \Pi_2 &\models T_2 \end{aligned}$$

These translation definitions can be combined in the following way to entail new theories

$$T_1 \cup \Delta_1 \cup \Pi_2 \models T_2' \quad T_2 \cup \Delta_2 \cup \Pi_1 \models T_1'$$

where T_1, T_1' are in the same hierarchy, and where T_2, T_2' are in the same hierarchy. We can therefore show that T_1 and T_2' are synonymous, and T_2 and T_1' are synonymous. The relationship between the upper ontologies T_1 and T_2 is now equivalent to the relationship between T_1 and T_1' on the one hand, and the relationship between T_2 and T_2' on the other.

Using the results from the previous section, the relationship of TUpperWare to other ontologies can easily be determined, since the mappings to DOLCE, SUMO are compositions of the mappings used in the verification of the generic ontologies. From translation definitions for SUMO-Time and the Process Specification Language (PSL)¹⁴, we can use this technique to generate the translation definitions directly between theories in the SUMO-Time and PSL Hierarchies:

$$(\forall x) \textit{TimePoint}(x) \equiv \textit{timepoint}(x) \tag{1}$$

$$(\forall x) \textit{TimeInterval}(x) \equiv \textit{timeinterval}(x) \tag{2}$$

$$(\forall x, y) \textit{begin}(x, y) \equiv (\textit{beginof}(x) = y) \tag{3}$$

$$(\forall x, y) \textit{end}(x, y) \equiv (\textit{endof}(x) = y) \tag{4}$$

$$(\forall x, y, z) \textit{interval}(x, y, z) \equiv (\textit{between}(x, y) = z) \tag{5}$$

Figure 1 summarizes the results with the verification of generic time ontologies and time modules within upper ontologies. Figure 1(a) shows the modules that axiomatize the ordering over timepoints, as well as the relationship between time intervals and timepoints. Figure 1(b) shows the mereology of time intervals that is axiomatized in the upper ontologies. Note that DOLCE only axiomatizes the mereology over time intervals, and does not contain a distinction between time intervals and timepoints, and hence does not specify an ordering over temporal elements.

7. Summary

The “sideways” perspective, in which upper ontologies are composed of generic ontologies, has been proposed as a way of more effectively using upper ontologies to support semantic integration and ontology design. This approach has been successfully applied to the verification and modularization of existing upper ontologies (SUMO and DOLCE). It is also being utilized in the design of a new upper ontology, TUpperWare, with the objective of semantically integrating a set of international standards. The design of TUpperWare addresses important research questions for upper ontologies:

¹⁴http://colore.oor.net/psl_core

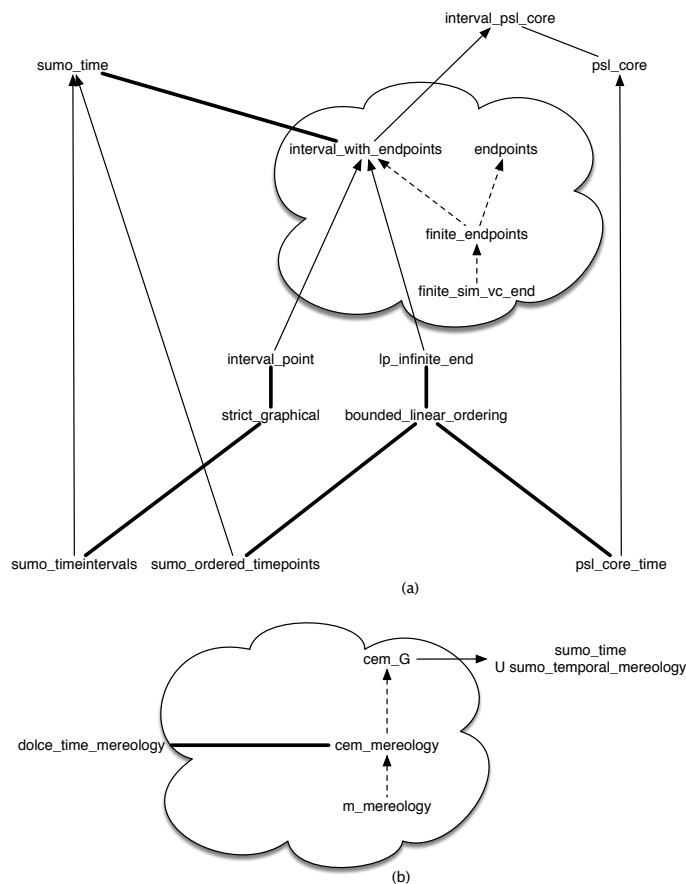


Figure 1. (a) Relationships between the time modules within upper ontologies. (b) Relationships between the mereology modules within upper ontologies. In both, cloud shapes denote hierarchies, dotted lines denote nonconservative extension, thin solid lines denote conservative extension, and thick lines denote synonymy.

Question 1 *In addition to the generic ontologies in Table 1, which ones are missing?*

For example, there are no modules of TUpperWare that are comparable to the notion of dependence in DOLCE.

Question 2 *Since upper ontologies are not complete logical theories, how incomplete can an upper ontology be without being trivial?*

This is not the question of whether the upper ontology has unintended models, since this can usually be addressed by extension with additional axioms. Instead, this question is about whether or not the models of the ontology should even be intended. On the other hand, if an upper ontology is essentially a taxonomy with few additional axioms, there will be many models, and it is difficult to claim that *all* of them will be intended.

References

- [1] Bahar AAmeri. Using Partial Automorphisms to Design Process Ontologies. In *FOIS*, pages 309–322, 2012.
- [2] Carmen Chui. Axiomatized Relationships between Ontologies. Master’s thesis, University of Toronto, November 2013.
- [3] Carmen Chui and Michael Grüninger. Mathematical Foundations for Participation Ontologies. In *Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS 2014, September, 22-25, 2014, Rio de Janeiro, Brazil*, pages 105–118, 2014.
- [4] Carmen Chui and Michael Grüninger. Merging the DOLCE and PSL Upper Ontologies. In *KEOD 2014 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, Rome, Italy, 21-24 October, 2014*, pages 16–26, 2014.
- [5] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching, Second Edition*. Springer, 2013.
- [6] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. Sweetening Ontologies with DOLCE. In *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 13th International Conference, EKAW 2002, Sigüenza, Spain, October 1-4, 2002, Proceedings*, pages 166–181, 2002.
- [7] Michael Grüninger, Torsten Hahmann, Ali Hashemi, Darren Ong, and Atalay Özgövde. Modular First-Order Ontologies via Repositories. *Applied Ontology*, 7(2):169–209, 2012.
- [8] Michael Grüninger, Torsten Hahmann, Megan Katsumi, and Carmen Chui. A sideways look at upper ontologies. In *Formal Ontology in Information Systems - Proceedings of the Eighth International Conference, FOIS 2014, September, 22-25, 2014, Rio de Janeiro, Brazil*, pages 9–22, 2014.
- [9] Oliver Kutz and Till Mossakowski. A Modular Consistency Proof for DOLCE. In *Proceedings of the Twenty-Fifth AAI Conference on Artificial Intelligence, AAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.
- [10] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. WonderWeb Deliverable D18 Ontology Library (Final). Technical report, IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web, 2003.
- [11] Lydia Silva Muñoz and Michael Grüninger. Mapping and verification of the time ontology in SUMO. In *Formal Ontology in Information Systems - Proceedings of the 9th International Conference, FOIS 2016, Annecy, France, July 6-9, 2016*, pages 109–122, 2016.
- [12] Lydia Silva Muñoz and Michael Grüninger. Verifying and Mapping the Mereotopology of Upper-Level Ontologies. In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 2: KEOD, Porto - Portugal, November 9 - 11, 2016.*, pages 31–42, 2016.
- [13] David Pearce and Agustín Valverde. Synonymous theories and knowledge representations in answer set programming. *J. Comput. Syst. Sci.*, 78(1):86–104, 2012.
- [14] A. Pease, I. Niles, and J. Li. The suggested upper merged ontology: A large ontology for the semantic web and its applications. In A. Pease, editor, *Ontologies and the Semantic Web, Papers from the AAI Workshop*. AAAI Press, 2002.
- [15] Lydia Silva-Munoz and Michael Grüninger. *Lecture Notes in Communications in Computer and Information Science*, chapter The Mereologies of Upper Ontologies. Springer, 2017.