
Workshop Notes

**The ECAI-10 Workshop on
Automated Reasoning about Context and Ontology Evolution**

ARCOE-10

**August 16-17, 2010
Lisbon, Portugal**

held at the
European Conference on Artificial Intelligence

Co-Chairs

**Alan Bundy
Jos Lehmann
Guilin Qi
Ivan José Varzinczak**

<http://www.arcoe.org>

Automated Reasoning about Context and Ontology Evolution

Alan Bundy^a, Jos Lehmann^a, Guilin Qi^b, Ivan José Varzinczak^c

^a School of Informatics, University of Edinburgh

^b School of Computer Science and Engineering, Southeast University

^c Meraka Institute, Pretoria, South Africa

Methods of automated reasoning have solved a large number of problems in Computer Science by using formal ontologies expressed in logic. Over the years, though, each problem or class of problems has required a different ontology, and sometimes a different version of logic. Moreover, the processes of conceiving, controlling and maintaining an ontology and its versions have turned out to be inherently complex. All this has motivated much investigation in a wide range of disparate disciplines - from logic-based Knowledge Representation and Reasoning to Software Engineering, from Databases to Multimedia - about how to relate ontologies to one another.

Just like the previous edition, ARCOE-10 aims at bringing together researchers and practitioners from core areas of Artificial Intelligence (Knowledge Representation and Reasoning, Contexts, and Ontologies) to discuss these kinds of problems and relevant results.

Historically, there have been at least three different, yet interdependent motivations behind this type of research: defining the relationship between an ontology and its context, providing support to ontology engineers, enhancing problem solving and communication for software agents.

ARCOE Call for Abstracts has been formulated against such historical background. Submissions to ARCOE-10 have been reviewed by two to three Chairs or PC members and ranked on relevance and quality. Approximately eighty percent of the submissions have been selected for presentation at the workshop and for inclusion in these Workshop Notes.

Thanks to the invaluable and much appreciated contributions of its Program Committee, its Invited Speakers and its authors, ARCOE-10 provides participants with an opportunity to position various approaches with respect to one another. Hopefully, though the workshop and these Notes will also start a process of cross-pollination and set out the constitution of a truly interdisciplinary research-community dedicated to automated reasoning about contexts and ontology evolution.

(Edinburgh, Nanjing, Pretoria – June 2010)

ARCOE-10 Co-Chairs

Alan Bundy (University of Edinburgh, UK)

Jos Lehmann (University of Edinburgh, UK)

Guilin Qi (Southeast University, China)

Ivan José Varzinczak (Meraka Institute, South Africa)

ARCOE-10 Invited Speakers

Tommie Meyer (Meraka Institute, South Africa)

Fausto Giunchiglia (University of Trento, Italy)

ARCOE-10 Program Committee

Grigoris Antoniou (FORTH, Greece)

Franz Baader (TU Dresden, Germany)

Richard Booth (Máhāa Wítáyaaalai Mahasarakham, Thailand)

Paolo Bouquet (Università di Trento, Italy)

Jerome Euzenat (INRIA Grenoble Rhone-Alpes, France)

Giorgos Flouris (FORTH, Greece)

Chiara Ghidini (FBK Fondazione Bruno Kessler, Italy)
Deborah McGuinness (Rensselaer Polytechnic Institute, USA)
Thomas Meyer (Meraka Institute, South Africa)
Maurice Pagnucco (The University of New South Wales, Australia)
Valeria de Paiva (Palo Alto Research Center (PARC), USA)
Jeff Pan (University of Aberdeen, UK)
Dimitris Plexousakis FORTH, Greece)
Luciano Serafini (FBK Fondazione Bruno Kessler, Italy)
Renata Wassermann (Universidade de São Paulo, Brazil)

Table of Contents

Marcio Ribeiro and Renata Wassermann <i>More about AGM Revision in Description Logic</i>	7-8
Richard Wallace and Tabbasum Naz <i>Context and Intention in Ontologies</i>	9-10
Christoph Benzmueller and Adam Pease <i>Reasoning with Embedded Formulas and Modalities in SUMO</i>	11-12
Hai Nguyen, Natasha Alechina and Brian Logan <i>Ontology Debugging with Truth Maintenance Systems</i>	13-14
Jos Lehmann, Alan Bundy and Michael Chan <i>Qualitative Causal Analysis of Empirical Knowledge for Ontology Evolution in Physi</i>	15-16
Claudia d'Amato and Nicola Fanizzi <i>Uncertainty Reasoning through Similarity in Context</i>	17-18
Adam Pease, Chris Benzmueller <i>Ontology Archaeology: Mining a Decade of Effort on the Suggested Upper Merged Ontology</i>	19-20
Thomas Meyer, Kodylan Moodley and Ivan Varzinczak <i>First Steps in the Computation of Root Justifications</i>	21-22
Michael Chan, Jos Lehmann and Alan Bundy <i>A Contextual Approach to Detection of Conflicting Ontologies</i>	23-24

More About AGM Revision in Description Logics

Márcio Moretto Ribeiro and Renata Wassermann
University of São Paulo
{marciomr, renata}@ime.usp.br

1 Introduction

Belief revision studies the dynamics of beliefs defining some operations in logically closed sets (belief sets): expansion, revision and contraction. Revision, in particular deals with the problem of accommodating consistently a newly received piece of information.

Most of the works on belief revision following the seminal paper [1] assume that the underlying logic of the agent satisfies some assumptions. In [5] we showed how to apply revision of belief sets to logics that are not closed under negation. We have, however, assumed that the logic satisfies a property called distributivity. In the present work we show a list of description logics that are not closed under negation and study which of them are distributive.

1.1 AGM paradigm

The most influential work in belief revision is [1]. In this work the authors defined a number of *rationality postulates* for contraction and revision, now known as the AGM postulates. The authors then showed *constructions* for these operations and proved that the constructions are equivalent to the postulates (*representation theorem*)

Most works in belief revision assume some properties on the underlying logic: *compactness*, *Tarskianicity*, *deduction* and *supraclassicality*, which we will refer to as the AGM assumptions. The last two together are equivalent to the following two properties together for Tarskian logics:

Definition 1 (distributivity) A logic $\langle \mathcal{L}, Cn \rangle$ is distributive iff for all sets of formulas $X, Y, W \in 2^{\mathcal{L}}$, we have that $Cn(X \cup (Cn(Y) \cap Cn(W))) = Cn(X \cup Y) \cap Cn(X \cup W)$.

Definition 2 (closure under negation) A logic $\langle \mathcal{L}, Cn \rangle$ is closed under negation iff for all $A \in 2^{\mathcal{L}}$ there is a $B \in 2^{\mathcal{L}}$ such that $Cn(A \cup B) = \mathcal{L}$ and $Cn(A) \cap Cn(B) = Cn(\emptyset)$. The set B is then called a negation of A .

AGM revision in non-classical logics: In [5] we argued that some description logics are not closed under negation and, hence, do not satisfy the AGM assumptions. Furthermore, the most common way to define revision is via Levi identity ($K * \alpha = K - \neg \alpha + \alpha$), which assumes the existence of the negation of α . We proposed then a new construction and a set of postulates for revision for logics that are not closed under negation.

We used two postulates, borrowed from the belief base literature:

(relevance) If $\beta \in K \setminus K * \alpha$ then there is K' such that $K \cap (K * \alpha) \subseteq K' \subseteq K$ and $K' \cup \{\alpha\}$ is consistent, but $K' \cup \{\alpha, \beta\}$ is inconsistent.

(uniformity) If for all $K' \subseteq K$, $K' \cup \{\alpha\}$ is inconsistent iff $K' \cup \{\beta\}$ is inconsistent then $K \cap K * \alpha = K \cap K * \beta$

The set of rationality postulates we considered is: closure, success, inclusion, consistency, relevance and uniformity. The following proposition is an evidence that this is a good choice of rationality postulates:

Proposition 3 For logics that satisfy the AGM assumptions, closure, success, inclusion, consistency, relevance and uniformity are equivalent to the original AGM postulates for revision: closure, success, consistency, vacuity and extensionality.

We proposed also a construction inspired in some ideas from [4]:

Definition 4 (Maximally consistent set w.r.t α) [4] $X \in K \downarrow \alpha$ iff $X \subseteq K$, $X \cup \{\alpha\}$ is consistent and if $X \subset X' \subseteq K$ then $X' \cup \{\alpha\}$ is inconsistent.

Definition 5 (Selection function) [1] A selection function for K is a function γ such that if $K \downarrow \alpha \neq \emptyset$, then $\emptyset \neq \gamma(K \downarrow \alpha) \subseteq K \downarrow \alpha$. Otherwise, $\gamma(K \downarrow \alpha) = \{K\}$.

The construction of a revision without negation is defined as $K *_{\gamma} \alpha = \bigcap \gamma(K \downarrow \alpha) + \alpha$.

We proved that, for distributive logics, this construction is completely characterized by the set of rationality postulates we are considering i.e. we proved the representation theorem relating the construction to the set of postulates [5].

1.2 Description Logics

Description logics (DLs) forms a family of formalisms to represent terminological knowledge. The *signature* of a description logic is a tuple $\langle N_C, N_R, N_I \rangle$ of concept names, roles names and individual names of the language [2]. From a signature it is possible to define *complex concepts* via a *description language*. Each DL has its own description language that admits a certain set of constructors.

The *semantic* of a DL is defined using an *interpretation* $\mathcal{I} = \langle \cdot^{\mathcal{I}}, \Delta^{\mathcal{I}} \rangle$ such that $\Delta^{\mathcal{I}}$ is a non-empty set called *domain* and $\cdot^{\mathcal{I}}$ is an *interpretation function*. For each concept name the interpretation associates a subset of the domain, for each role name a binary relation in the domain and for each individual an element of the domain. The interpretation is then extended to complex concepts.

A *sentence* in a DL is a restriction to the interpretation. A *TBox* is a set of sentences of the form $C_1 \sqsubseteq C_2$ that restricts the interpretation of concepts¹, an *ABox* is a set of sentences of the form $C(a)$, $R(a, b)$,

¹ Assuming that the logic admits GCI axioms

$a = b$ and $a \neq b$ that restricts the interpretation of individuals. Some DLs, like $\mathcal{ALCH}$, admits also an *RBox* which is a set of sentences of the form $R \sqsubseteq S$ that restricts the interpretations of roles.

Let $\Sigma = \langle \mathcal{T}, \mathcal{A}, \mathcal{R} \rangle$ be a tuple where $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{R}$ are a Tbox, an ABox and an RBox respectively. A sentence α is a *consequence* of Σ ($\Sigma \models \alpha$ or $\alpha \in Cn(\Sigma)$) iff for all interpretations $\mathcal{I}$ if $\mathcal{I}$ satisfies Σ then $\mathcal{I}$ satisfies α .

Two characteristics the DLs we are considering that will be important in this work are: in $\mathcal{ALC}$ every sentence in the TBox is equivalent to a sentence of the form $\top \sqsubseteq C$ for a concept C [3] and in $\mathcal{ALCO}$ every sentence in the ABox is equivalent to a sentence of the form $\top \sqsubseteq C$.

2 Properties of Description Logics

The main contribution of this work is to show a set of description logics that are not closed under negation and which of them are distributive i.e. we show a set of logics such that representation theorem for revision without negation is applicable. It turns out that most DLs that admits GCI axioms (GCI axioms allow complex concepts in both sides of the sentence) are not closed under negation, but many of them are also not distributive.

Classic negation in DLs: We will say that two roles R and S are *unrelated* iff neither $R \sqsubseteq S \in Cn(\emptyset)$ nor $S \sqsubseteq R \in Cn(\emptyset)$. The main result of this section proves that if the signature of a DL $\langle \mathcal{L}, Cn \rangle$ has infinitely many unrelated roles and admits $\forall, \sqcup, \neg$ and GCI axioms then $\langle \mathcal{L}, Cn \rangle$ is not closed under negation.

Theorem 6 Consider a DL $\langle \mathcal{L}, Cn \rangle$ that admits the constructors $\neg, \forall, \sqcup$ and general concept inclusion axioms in the TBox. If there is an infinite number of unrelated roles, then $\langle \mathcal{L}, Cn \rangle$ is not closed under negation

The proof of this theorem comes from the fact that if $\langle \mathcal{L}, Cn \rangle$ admits $\sqcup, \neg$ then every sentence can be written as $\top \sqsubseteq C$ and from the following lemmas:

Lemma 7 Let A and B be concepts such that $\top \sqsubseteq A$ and $\top \sqsubseteq B$ are not tautologies and let R be a role name that is unrelated with any role that appears in A or B . Then $Cn(\emptyset) \subset Cn(\top \sqsubseteq A \sqcap \forall R.B) \subseteq Cn(\top \sqsubseteq A) \cap Cn(\top \sqsubseteq B)$.

Lemma 8 If $Cn(\top \sqsubseteq A) = Cn(\emptyset)$ and $\top \sqsubseteq B$ is a negation of $\top \sqsubseteq A$ then $Cn(\top \sqsubseteq B) = \mathcal{L}$

As a corollary of this result we have that many well known description logics are not closed under negation. Hence, for all these logics the AGM results are not applicable:

Corollary 9 The following DLs are not closed under negation: $\mathcal{ALC}$, $\mathcal{ALCO}$, $\mathcal{ALCH}$, OWL-lite and OWL-DL.

Distributivity in DLs: In this section we show a list of distributive and non-distributive DLs. We start with an example showing that the logic $\mathcal{ALC}$ is not distributive in general.

Example 10: Let $X = \{a = b\}$, $Y = \{C(a)\}$ and $Z = \{C(b)\}$, then $Cn(Y) \cap Cn(Z) = Cn(\emptyset)$. Hence $C(a) \notin Cn(X \cup (Cn(Y) \cap Cn(Z)))$, but $C(a) \in Cn(X \cup Y) \cap Cn(X \cup Z)$.

The example above depends on the existence of the ABox. In fact, $\mathcal{ALC}$ with empty ABox is distributive:

Proposition 11 Consider a DL $\langle \mathcal{L}, Cn \rangle$ such that for every sentence $\alpha \in \mathcal{L}$ there is a sentence $\alpha' \in \mathcal{L}$ such that $Cn(\alpha) = Cn(\alpha')$ and α' has the form $\top \sqsubseteq C$ for some concept C . Then $\langle \mathcal{L}, Cn \rangle$ is distributive.

Since in $\mathcal{ALCO}$ the ABox can be written in terms of the TBox, $\mathcal{ALCO}$ is distributive even in the presence of the ABox.

Finally, if we consider a logic $\langle \mathcal{L}, Cn \rangle$ that admits role hierarchy, but does not admit role constructors, then $\langle \mathcal{L}, Cn \rangle$ is not distributive. Consider the following example:

Example 12: Let $X = \{R \sqsubseteq S_1, R \sqsubseteq S_2\}$, $Y = \{S_1 \sqsubseteq S_3\}$ and $Z = \{S_2 \sqsubseteq S_3\}$. We have that $Cn(Y) \cap Cn(Z) = Cn(\emptyset)$. Hence $R \sqsubseteq S_3 \notin Cn(X \cup (Cn(Y) \cap Cn(Z)))$, but $R \sqsubseteq S_3 \in Cn(X \cup Y) \cap Cn(X \cup Z)$.

Besides $\mathcal{ALCH}$, the logics behind OWL 1 (*SHOIN* for OWL-DL and *SHIF* for OWL-lite), OWL-2 (*SROIQ*) and the OWL profiles OWL-RL and OWL-QL admit role hierarchy, but do not admit role constructors. None of these logics are distributive.

The following table sums up the results of this section:

Description Logic	Negation	Distributivity
$\mathcal{ALC}$	no	no
$\mathcal{ALC}$ without ABox	no	yes
$\mathcal{ALCO}$	no	yes
$\mathcal{ALCH}$, OWL-lite, OWL-DL	no	no
OWL-QL, OWL-RL and OWL 2	?	no

3 Conclusion and future work

In this work we continued the work started in [5] by showing for which DLs the AGM revision without negation can be applied. We showed that most DLs that admits GCIs are not closed under negation, but most of them are also not distributive. We showed that $\mathcal{ALC}$ with empty TBox and $\mathcal{ALCO}$ are two exceptions. These logics are distributive and not closed under negation. Hence, the representation theorem presented in [5] holds for $\mathcal{ALC}$ with empty ABox and $\mathcal{ALCO}$.

In addition to that, we showed that the postulates used in [5] are equivalent to the AGM postulates if the underlying logic satisfies the AGM assumptions. This is a good evidence that we chose a good set of rationality postulates.

As future work we should look for a construction that can be characterized by this set of postulates (or a similar one) not only in distributive, but in any Tarskian compact logic.

REFERENCES

- [1] C. Alchourrón, P. Gärdenfors, and D. Makinson, 'On the logic of theory change', *Journal of Symbolic Logic*, **50**(2), 510–530, (1985).
- [2] *The Description Logic Handbook*, eds., F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, Cambridge University Press, 2003.
- [3] F. Baader and U. Sattler, 'An overview of tableau algorithms for description logics', *Studia Logica*, **69**(1), 5–40, (2001).
- [4] J. P. Delgrande, 'Horn clause belief change: Contraction functions', in *Proceedings of KR*, eds., G. Brewka and J. Lang, pp. 156–165, (2008).
- [5] M. M. Ribeiro and R. Wassermann, 'AGM revision in description logics', in *Proceedings of ARCOE*, (2009).

Context and Intention in Ontologies

Richard J. Wallace and Tabassum Naz¹

1 EXTENDED ABSTRACT

1.1 (Meta-)Context

Ontologies are only useful within a given context. Sometimes this context is quite specific; sometimes it is broad-based or generic.

In many cases, the context of an ontology can be described as the *intention* of that ontology. In fact, most practical ontologies are used for a specific purpose, and this *intentionality* is usually reflected throughout their organisation. This includes the concepts defined, the division of superordinate concepts into subordinate categories, and the properties that are specified.

However, more often than not this purpose remains implicit. It may not even be expressed in any of the concepts in the ontology. More significantly, the relevance of a given concept with respect to the purpose of the ontology is never specified in a clear, unambiguous fashion. Instead, concepts are apparently included (or excluded) on the basis of intuition and trial-and-error.

More generally, the implications of intentionality for ontology organisation are not clear and have never been spelled out.

1.2 An example

The first example is taken from the travel domain. Portions of two independently created ontologies are shown in Figure 1. The "Tourism" ontology (left panel in figure) was developed for Semantic Web sites related to tourism [4]. The "e-tourism" ontology, also known as "OnTour" (right-hand panel) is part of a Web assistant to aid users searching for vacation packages [7]. Despite the similarity of domain and intention, the ontologies are strikingly different. The concept hierarchies are quite dissimilar, as are the properties defined (not shown in figure).

1.3 Issues raised by this example

The tourism example shows how extensive differences in ontology organisation can be when the intention is (apparently) the same.

It should also be noted that in neither case is the purpose, or intention, of the ontology explicitly represented. Thus, there is no concept of a trip or of planning a trip. There is also an issue of the level or generality of a concept. For example the e-tourism ontology has <Ticket>, <Location>, <DateTime>, and <ContactData> at the same level in the ontology, but these concepts differ greatly in abstraction.

More importantly, these concepts have very different relations to the activity of aiding tourists or searching for vacation packages. Some, like <Location> and <DateTime>, are general concepts associated with basic 'stage setting', while others like <Ticket>,



Figure 1. Portions of two ontologies developed independently for the travel domain.

<Event>, and <ContactData> are associated with specific roles in the activity that the ontology is meant to support.

In general, when perusing these ontologies, one has the sense of an overall lack of coherence, although at present it is difficult to specify what form this coherence should take. This, in fact, is the problem that the present research will attempt to address.

1.4 Existing and other possible approaches

- Meta-ontology of intentions: given ontology as an individual belonging to some class of intentions.

Example: In the Ontology Metadata Vocabulary (OMV)[5], the concept <OntologyTask> contains information about the task the ontology was intended to be used for. <OntologyTask> has further pre-defined tasks i.e. <AnnotationTask>, <MatchingTask>, <IntegrationTask>, <QueryFormulationTask>, etc.

(In our view, there is something awkward and inept about this approach, as if the machinery of ontologies was being taken off the shelf and put to use in a rote, unthinking fashion.)

- Intentionality might be specified by means of top-level concepts. This might involve a database-view approach, based on the class/subclass relations in the full ontology.

1.5 Present approach

We begin with the question: Is intention a well-defined concept? So that dealing with it is a well-defined problem? In fact, there is a literature of some proportions in philosophy that deals with this question.

¹ Cork Constraint Computation Centre, University College Cork, Cork, Ireland, email: r.wallace@4c.ucc.ie, t.naz2001@yahoo.com

In particular, the work of Bratman [1] includes an extensive discussion of this topic. And it is his definition that we will use. Therefore, we consider the idea of intention as bound up with the creation of a plan.

Thus, we approach ontology construction as if we were building a plan. This means that we must start by defining the goal. The goal will be one of the concepts in the ontology.

It seems most natural to use the HTN style of planning [9], in which a major action is plan decomposition. An example is shown below for the tourism domain. Here, the goal is to support trip-planning. This goal is then decomposed into subgoals; alternatively, we can think of a basic action trip-planning, decomposed into component actions. (Note that <trip> is not included in either of the ontologies cited above, although it is a key concept in this application domain.)

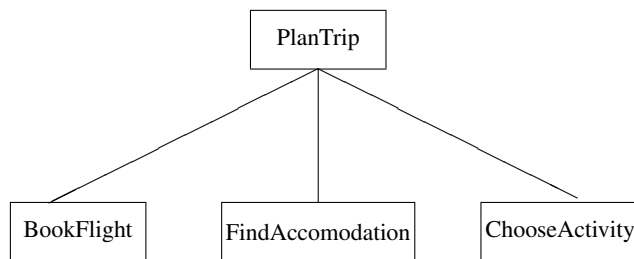


Figure 2. Top level of plan for tourism ontology.

In each case, we assume a mapping between an action and a concept. We also map in the same way between preconditions as well as from effects to concepts. This is basically how we build our ontology.

This may also give us a way of evaluating the completeness of the ontology. That is, an ontology is complete only if it is associated with a set of actions, etc. that can accomplish the basic goal, which represents the intention of the ontology.

Many details remain to be worked out, e.g. how to guide the user in the planning process, how to update and revise the growing ontology, the relation, if any, of the plan structure to the structure of the ontology, and how to set up properties appropriately (the precondition-action-effect relations may help to guide this aspect of ontology building). We must also explore the various forms of abstraction in planning [8] to see if they have any bearing in this context.

As part of the process of elaborating, and checking, our ideas, we are building a system for ontology planning, tentatively called On-toPlanner. Currently, the basic scheme of operation that we envisage is to (i) parse the predicates used in plan construction to obtain nouns and verbs which indicate relevant concepts, (ii) characterise or 'locate' the concepts via a top-level ontology, (iii) introduce these concepts and possibly related concepts and properties into the developing ontology.

1.6 Further issues

- Most ontologies incorporate more than one intention - usually necessary (just as an automobile or a mobile phone incorporates more than one intention).
- Intentionality has implications for merging ontologies, again which have not been worked out. Merging and matching might well be facilitated if the intentionality of each ontology had an explicit representation.

- Characterising an ontology with respect to intentions. Here, a possible approach is to create a partial order based on intentions, i.e. lattice structure, in order to locate an ontology within a lattice of ontologies. The supremum might be all the intentions for that lattice.
- Historical aspects of ontologies, i.e. developing/emerging intentions.
- Intentions and agents (esp. BDI agents). Explicit intentions may enhance the accessibility of an ontology to software agents.

1.7 Unrelated Work.

There are many papers unrelated to the present work. Here, our concern is with those cases where this might not be recognised. An obvious example is the development of ontologies that are meant to be used in connection with planning. This topic has received a fair amount of attention during the past few years. In fact, a recent ICAPS workshop was devoted to this topic [2]. More recently, Jingge et al. have discussed how to combine ontologies with HTN planning [3]. It should be obvious that this work has little or nothing in common with the present work, whose purpose is to guide ontology building rather than to provide a planner with domain knowledge.

A second type of unrelated work is concerned with developing ontologies to characterise the general planning process. Here, the only example that we are aware of is a paper by Rajpathak and Motta [6]. Although this seems less distantly related to our concerns than the works cited above, it should be clear that the purpose of this ontology, which is to aid planning at a generic level, is not what we have described in this paper. Nonetheless, although the intention is quite different in the two cases, an ontology of plans may be useful in the present context.

ACKNOWLEDGEMENTS

This work was supported by Science Foundation Ireland under Grant 05/IN/I886. and by the Irish Marine Institute, Project Ref. No. isPBA/KI/07/001.

REFERENCES

- [1] M. E. Bratman, *Intention, Plans, and Practical Reason*, Harvard, 1987.
- [2] ICAPS05, *ICAPS 2005 Workshop on the Role of Ontologies in AI Planning and Scheduling*, 2005.
- [3] S. Jingge, C. Jianzhong, and L. Yiping, 'Combining ontology engineering with htn planning', in *Proc. 2006 Conference on Leading the Web in Concurrent Engineering: Next Generation Concurrent Engineering*, pp. 214–221. IOS, (2006).
- [4] T. Kalinka, *Tourism.owl*, http://protegewiki.stanford.edu/index.php/Protege_Ontology_Library#OWL_ontologies.
- [5] R. Palma, J. Hartmann, and P. Haase, *OMV – Ontology Metadata Vocabulary for the Semantic Web*, Technical Report Version 2.4.1, 2009.
- [6] D. Rajpathak and E. Motta, 'An ontological formalization of the planning task', in *Proc. Third International Conference on Formal Ontology in Information Systems - FOIS 2004*, eds., A. Varzi and L. Vieu, pp. 305–316. IOS, (2004).
- [7] K. Siorpass, K. Prantner, and D. Bachlechner, *e-tourism Ontology*, Digital Enterprise Research Institute, Galway, <http://e-tourism.deri.at/ont/index.html>, 2005.
- [8] J. D. Tenenbergh, 'Abstraction in planning', in *Reasoning About Plans*, eds., J. F. Allen, H. A. Kautz, R. N. Pelavin, and J. D. Tenenbergh, 213–284, Morgan Kaufmann, (1991).
- [9] D. Wilkins, *Practical Planning*, Morgan Kaufmann, 1988.

Reasoning with Embedded Formulas and Modalities in SUMO¹

Benzmüller Christoph and Pease Adam²

Abstract. Reasoning with embedded formulas is relevant for the SUMO ontology but there is limited automation support so far. We investigate whether higher-order automated theorem provers are applicable for the task. Moreover, we point to a challenge that we have revealed as part of our experiments: modal operators in SUMO are in conflict with Boolean extensionality. A solution is proposed.

1 EMBEDDED FORMULAS IN SUMO

The open source Suggested Upper Merged Ontology³ (SUMO) [9] (and similarly, proprietary Cyc [13]) contains a small but significant amount of higher-order representations. The approach taken in these systems to address higher-order challenges has been to employ specific translation ‘tricks’, possibly in combination or in addition to some pre-processing techniques. Examples of such means are the quoting techniques for embedded formulas as employed in SUMO [11] and the heuristic-level modules in CYC [13]. Unfortunately, however, these solutions are strongly limited. The effect is that many desirable inferences are currently not supported, so that many relevant queries cannot be answered.

This includes statements in which formulas are embedded as arguments of terms, for example, statements that employ epistemic operators such as *believes* or *knows*, temporal operators such as *holdsDuring*, and further operators such as *disapproves* or *hasPurpose*. While first-order automated theorem proving (FO-ATP) for SUMO has strongly improved recently [12], there is still only very limited support for reasoning with non-trivial embedded formulas; we give an example (free variables in premises are universal and those in the query are existential):

Ex. 1 (Reasoning in temporal contexts.) *What holds that holds at all times. Mary likes Bill.⁴ During 2009 Sue liked whoever Mary liked. Is there a year in which Sue has liked somebody?*

A: $(\Rightarrow ?P \text{ (holdsDuring ?Y ?P)})$
B: $(lk \text{ Mary Bill})$
C: $(holdsDuring \text{ (YearFn 2009)}$
 $\text{ (forall (?X) } (\Rightarrow (lk \text{ Mary ?X}) (lk \text{ Sue ?X}))))$
Q: $(holdsDuring \text{ (YearFn ?Y) (lk Sue ?X)})$

This example, which is a challenge for FO-ATP (note the embedded first-order formula), is actually trivial for higher-order automated theorem provers (HO-ATP): the prover LEO-II [5] can solve it in 0.16

sec. on a standard MacBook. A slight modification of Ex.1, which LEO-II proves in 0.08 sec., is:⁵

Ex. 2 (Ex.1 modified; A is replaced by ‘True always holds’.)

A’: $(holdsDuring \text{ ?Y True})$
B: $(lk \text{ Mary Bill})$
C: $(holdsDuring \text{ (YearFn 2009)}$
 $\text{ (forall (?X) } (\Rightarrow (lk \text{ Mary ?X}) (lk \text{ Sue ?X}))))$
Q: $(holdsDuring \text{ (YearFn ?Y) (lk Sue ?X)})$

Further examples are studied in [6]; there we also outline the translation from SUMO’s SUO-KIF representation language [10, 7] as used above to the new higher-order TPTP THF syntax [14] as supported by several HO-ATPs including LEO-II.

2 THE PROBLEM WITH MODAL OPERATORS

Validity of Ex.1 and Ex.2 is easily shown provided that Boolean extensionality⁶ is assumed (this ensures that the denotation of each formula, also the embedded ones, is either true or false). This assumption has actually never been questioned for SUMO, neither in [7] nor in [10].

However, this assumption also leads to problematic effects as the following slight modification of Ex.2 illustrates:

Ex. 3 (Ex.2 modified; now formulated for an epistemic context)

A’: $(knows \text{ ?Y True})$
B: $(lk \text{ Mary Bill})$
C’: $(knows \text{ Chris}$
 $\text{ (forall (?X) } (\Rightarrow (lk \text{ Mary ?X}) (lk \text{ Sue ?X}))))$
Q’: $(knows \text{ Chris (lk Sue Bill)})$

Using Boolean extensionality the query is easily shown valid and LEO-II can prove it in 0.04 sec. However, now this inference is disturbing since we have not explicitly required that $(knows \text{ Chris (lk Mary Bill)})$ holds which intuitively seems mandatory. Hence, we here (re-)discover an issue that some logicians possibly claim as widely known: modalities have to be treated with great care in classical, extensional higher-order logic. Our ongoing work therefore studies how we can suitably adapt the modeling of affected modalities in SUMO in order to appropriately address this issue. A respective proposal is sketched next.

¹ This work is funded by the German Research Foundation under grant BE 2501/6-1.

² Articulate Software, email: cbenzmueeller@apease@articulatesoftware.com

³ SUMO is available at <http://www.ontologyportal.org>

⁴ To save space ‘likes’ is written as ‘lk’.

⁵ It is important to note that *True* in A’ can actually be replaced by other tautologies, e.g. by $(equal \text{ Mary Mary})$; this may appear more natural and the example can still be proved by LEO-II in milliseconds.

⁶ For a detailed discussion of functional and Boolean extensionality in classical higher-order logic we refer to [2].

3 REASONING WITH MODALITIES IN SUMO

The solution we currently explore is to treat SUMO reasoning problems that involve modal operators as problems in quantified multimodal logics. Unfortunately there are only very few direct theorem provers for quantified multimodal logics available. We therefore exploit our recent embedding of quantified multimodal logics in classical higher-order logic [4, 3] and we investigate whether this embedding can fruitfully support the automation of modal operators in SUMO with off-the-shelf HO-ATPs (cf. [1] for first studies).

The idea of the embedding is simple: modal formulas are lifted to predicates over possible worlds, i.e. HO-terms of type $\iota \rightarrow o$, where ι is a reserved base type denoting the set of possible worlds. For individuals we reserve a second base type μ .⁷

Modal operators such as $\top$, $\perp$, $\neg$, $\vee$, $\supset$, $\Box$, and $\forall^{ind}$, $\forall^{prop}$ are then simply defined as abbreviations of proper HO-terms, e.g. $\neg = \lambda\phi_{\iota \rightarrow o} \lambda W_{\iota} \neg \phi W$ and $\Box = \lambda R_{\iota \rightarrow \iota \rightarrow o} \lambda\phi_{\iota \rightarrow o} \lambda W_{\iota} \forall V_{\iota} \neg R W V \vee \phi V$. In the following we write $\Box_r$ for the application of $\Box$ to an accessibility relation r of type $\iota \rightarrow \iota \rightarrow o$.⁸

Exploiting this embedding we can now suitably map SUMO problems containing modal operators: e.g. **C'** in Ex. 3 is lifted to $(\Box_{Chris} \forall^{ind} X_{\mu} ((lk Mary X) \supset (lk Sue X)))_{\iota \rightarrow o}$ and **B** simply becomes $(lk Mary Bill)_{\iota \rightarrow o}$. Since $\Box$ is here associated with knowledge we axiomatize it as an S4 or S5 modality below. Similarly, we could introduce further copies of $\Box$, e.g. for *believes*, and provide different axioms for it.

The final step is to ground the lifted terms. For this, T-Box like information in SUMO, such as the axiom (*instance instance BinaryPredicate*), is interpreted as universally quantified over all possible worlds: $\forall W_{\iota} ((instance instance BinaryPredicate) W)$. A-Box like information and queries in contrast are modeled with respect to a current world cw (of type ι). Since our examples only contain local premises and queries, i.e. A-Box like information, Ex.3 is thus translated as:

Ex. 4 (Translated Ex.3)

A'': $\forall Y_{\iota \rightarrow \iota \rightarrow o} ((\Box_Y \top) cw)$

B': $((lk Mary Bill) cw)$

C': $((\Box_{Chris} (\forall^{ind} X_{\mu} ((lk Mary X) \supset (lk Sue X)))) cw)$

Q': $((\Box_{Chris} (lk Sue Bill)) cw)$

The axioms for S4 (**T+4**) or S5 (**T+5**) can be added as follows:

T: $\forall W_{\iota}, Y_{\iota \rightarrow \iota \rightarrow o} ((\forall^{prop} \phi_{\iota \rightarrow o} \Box_Y \phi \supset \phi) W)$

4: $\forall W_{\iota}, Y_{\iota \rightarrow \iota \rightarrow o} ((\forall^{prop} \phi_{\iota \rightarrow o} \Box_Y \phi \supset \Box_Y \Box_Y \phi) W)$

5: $\forall W_{\iota}, Y_{\iota \rightarrow \iota \rightarrow o} ((\forall^{prop} \phi_{\iota \rightarrow o} \Box_Y \neg \phi \supset \Box_Y \neg \Box_Y \phi) W)$

The above example is not valid, which we wanted to achieve, and LEO-II correctly fails to prove it (timeout). However, if we move premise **B** in the context of Chris' knowledge then we get:

Ex. 5 (Modified Ex.4)

A'': $\forall Y_{\iota \rightarrow \iota \rightarrow o} ((\Box_Y \top) cw)$

B': $((\Box_{Chris} (lk Mary Bill)) cw)$

C': $((\Box_{Chris} (\forall^{ind} X_{\mu} ((lk Mary X) \supset (lk Sue X)))) cw)$

Q': $((\Box_{Chris} (lk Sue Bill)) cw)$

Ex.5 is valid and it is proved by LEO-II in less than 0.15 sec.

⁷ In [4] we use μ for possible worlds and ι for individuals; this syntactic switch is completely unproblematic.

⁸ Note the elegant way in which indexation over different accessibility relations is facilitated via λ -abstraction in the definition of $\Box$; this is the basis for combining different $\Box_k$ operators in our framework; cf. [1].

4 CONCLUSION

Reasoning with embedded formulas is naturally supported in extensional HO-ATPs. However, this leads to a problem regarding the adequate treatment of modal operators. A potential solution has been outlined in this paper that we are currently investigating further. Our ongoing work in particular studies the scalability of HO-ATPs for the task. Due to the recent, strong improvements of HO-ATPs [14] – which will be further fostered by the new higher-order CASC competitions – we are quite optimistic though. The large theories challenge obviously requires the development or adaptation of strong relevance filters, such as SInE [8], to our higher-order logic setting.

We have to admit that we currently see few alternatives to HO-ATP for the automation of ontology reasoning problems with embedded formulas and modalities as presented in this paper and already our toy examples seem challenging for other approaches.

Acknowledgment: We would like to thank Geoff Sutcliffe and Larry Paulson for their support and their contributions to our work.

REFERENCES

- [1] C. Benzmüller, 'Simple type theory as framework for combining logics', in *Context paper at the World Congress and School on Universal Logic III (UNIOLOG'2010)*, Lisbon, Portugal, (2010). <http://arxiv.org/abs/1004.5500>.
- [2] C. Benzmüller, C. Brown, and M. Kohlhasse, 'Higher-order semantics and extensionality', *J. of Symbolic Logic*, **69**(4), 1027–1088, (2004).
- [3] C. Benzmüller and L. C. Paulson, 'Multimodal and intuitionistic logics in simple type theory', *The Logic J. of the IGPL*, (2010).
- [4] C. Benzmüller and L. C. Paulson, *Quantified Multimodal Logics in Simple Type Theory*, SEKI Report SR-2009-02 (ISSN 1437-4447), SEKI Publications, DFKI Bremen GmbH, Safe and Secure Cognitive Systems, Cartesium, Enrique Schmidt Str. 5, D-28359 Bremen, Germany, 2009. <http://arxiv.org/abs/0905.2435>.
- [5] C. Benzmüller, L. C. Paulson, F. Theiss, and A. Fietzke, 'LEO-II — A Cooperative Automatic Theorem Prover for Higher-Order Logic', in *Proceedings of the 4th International Joint Conference on Automated Reasoning*, eds., P. Baumgartner, A. Armando, and D. Gilles, number 5195 in LNAI, pp. 162–170, Springer, (2008).
- [6] C. Benzmüller and A. Pease, 'Progress in automating higher-order ontology reasoning', in *Proceedings of the Second International Workshop on Practical Aspects of Automated Reasoning*, Edinburgh, UK, July 14, 2008, eds., B. Konev, R. A. Schmidt, and S. Schulz, CEUR Workshop Proceedings. CEUR-WS.org, (2010).
- [7] P. Hayes and C. Menzel, 'A semantics for the knowledge interchange format', in *IJCAI 2001 Workshop on the IEEE Standard Upper Ontology*, (2001).
- [8] K. Hoder, *Automated Reasoning in Large Knowledge Bases*, Master's thesis, Department of Theoretical Computer Science and Mathematical Logic, Charles University, Prague, 2008.
- [9] I. Niles and A. Pease, 'Towards a standard upper ontology', in 2nd International Conference on Formal Ontology in Information Systems (FOIS), eds., C. Welty and B. Smith, pp. 2–9, (2001). <http://www.adampease.org/professional/FOIS.pdf>.
- [10] A. Pease, Standard upper ontology knowledge interchange format. http://sigmakee.cvs.sourceforge.net/*checkout*/sigmakee/sigma/suo-kif.pdf.
- [11] A. Pease and G. Sutcliffe, 'First order reasoning on a large ontology', in *ESARLT*, eds., G. Sutcliffe, J. Urban, and S. Schulz, volume 257 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2007).
- [12] A. Pease, G. Sutcliffe, N. Siegel, and S. Trac, 'Large theory reasoning with SUMO at CASC', *AI Commun.*, **23**(2-3), 137–144, (2010).
- [13] D. Ramachandran, P. Reagan, and K. Goolsbey, 'First-ordered ResearchCyc: Expressivity and efficiency in a common-sense ontology', in *Papers from the AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*, Pittsburgh, Pennsylvania, (July 2005).
- [14] G. Sutcliffe and C. Benzmüller, 'Automated reasoning in higher-order logic using the TPTP THF infrastructure', *J. of Formalized Reasoning*, **3**(1), 1–27, (2010).

Ontology Debugging with Truth Maintenance Systems

Hai H. Nguyen, Natasha Alechina and Brian Logan¹

1 Introduction

An ontology is a description of a particular domain in terms of its concepts and relationships. Ontologies can be used by ‘intelligent agents’ to model their environment and communicate with other agents. The quality of ontologies are therefore crucial to the performance of intelligent agents. However, as with any other knowledge base, there is always the possibility that an ontology have some semantic defects. This abstract presents an approach to ontology debugging using ideas borrowed from Truth Maintenance Systems (TMS).

2 Ontology Debugging

This section briefly introduces the key problems in ontology debugging and briefly outlines some of the major work in the field. Firstly, we provide basic definitions of incoherence and inconsistency of a DL-based ontology.

Definition 1. An ontology is **incoherent** iff there is at least one unsatisfiable concept in its TBox.

Definition 2. An ontology is **inconsistent** iff there is no model for it.

Generally speaking, the incoherence problem deals with concept-unsatisfiability within the TBox while an inconsistency problem also involves assertional axioms.² *Ontology debugging* is the process of identifying bugs and producing repair plans for an incoherent or inconsistent ontology. Most work has been done in ontology debugging is for the incoherence problem (i.e., debugging and repair of unsatisfiable concepts) although recently the problem of inconsistency has also been investigated.

Basically, the process of debugging ontology has two parts. The first is to identify which sets of axioms (or parts of axioms) are responsible for an unsatisfiable concept or an inconsistency. The next step is to propose how can these axioms be modified to make the concept satisfiable, or to restore consistency to the Knowledge Base (KB) with respect to some particular criteria.

Two main approaches to pinpointing problematic axioms have been proposed in the literature: glass-box and black-box. Glass-box methods, e.g., [6, 7, 8, 9], use tableau-like rules to pinpoint the problematic axioms (concepts). These approaches obviously depend on a particular DL, as they have to modify the tableau rules to store and retrieve the sources of errors. Black-box methods on the other hand,

e.g., [6, 11, 4], are reasoner-independent, since they only use the reasoner as an external component to diagnose whether an a concept is satisfiable with respect to a particular T-Box (or KB in the case of inconsistency problem). There are also hybrid approaches, e.g., [5], which combine both glass box and black box approaches.

In this paper, the rules which we employ to create the dependency graph are similar to the classical tableau rules, as in some of the glass-box approaches, e.g., [7, 8].

3 Truth Maintenance Systems

Truth Maintenance Systems (TMS), e.g., [3], also known as Belief Revision Systems or Reason Maintenance Systems, play a central role in a style of belief revision called foundational belief revision. A TMS keeps track of dependencies between data to maintain the consistency of a database. A TMS consists of a set datum nodes and the justifications for them. A justification can be considered as a record of an inference, linking a datum node with the datum nodes used to derive it. Using these recorded dependencies, a TMS allows a problem-solver to quickly determine which nodes are “responsible” for belief in a particular datum.

According to [10], a TMS performs three main tasks: 1) given an assertion, find the assertions or assumptions used to derive it; 2) given a set of assumptions, find all the assertions can be derived from them; and 3) delete an assertion and all the consequences which have been derived from it. These tasks are also relevant to the problem of ontology debugging. For example, tracing the sources S_1 and S_2 of the assertions $A(x)$ and $\neg A(x)$, where A is a concept name and x is an individual in the ontology, gives the source of the contradiction (or clash) $S_1 \cup S_2$. Similarly, if one can find a minimal set of assumptions from which the contradictory assertions were derived, the minimal set of axioms which are the cause for the clash can also be identified.³ This set corresponds to a *MUPS* in [9], or a *justification* for concept unsatisfiability defined in [5].

4 Using ATMS for Ontology Debugging

One particular type of TMS is an Assumption-based TMS (ATMS) [1]. In an ATMS, each node is associated with the set of sets of assumptions used to derive it, as well as the datum nodes that constitute its immediate antecedents. These sets of assumptions are termed *environments*, and are always kept minimal and consistent. In this way, backtracking is avoided and multiple solutions can be found at the same time.

³ In the literature of ontology debugging, the idea of tagging an assertion with the axioms used to derive it has also been proposed in [7, 8].

¹ School of Computer Science, University of Nottingham UK, {hnn,nza,bsl}@cs.nott.ac.uk

² Note that the satisfiability checking problem in TBox can be reduced to a consistency checking problem by trying to construct a model for a concept using tableau rules.

In this section, we present an approach to ontology debugging using an ATMS. We focus on the problem of axiom pinpointing for an unsatisfiable concept (i.e., finding a set of axioms responsible for an unsatisfiable concept), and for simplicity, we only consider the unfoldable $\mathcal{ALC}$ TBOX without disjunctions.⁴ We show how the ATMS can be used to detect contradictions and to pinpoint sets of problematic axioms.

As a reasoner can easily detect that a concept is unsatisfiable by a satisfiability check, the key problem is to identify the sources of the unsatisfiability. This is the task of the ATMS. An ATMS node N_{datum} is of the form: $\langle datum, label, justifications \rangle$, where $datum$ is an assertion such as $A_i(a)$, $label$ is a set of environments (explained below), and $justifications$ are the sets of nodes that directly derive N_{datum} . Since there are many ways a datum can be derived, it is possible to have multiple justifications for a particular node. The ATMS distinguishes two special types of datum nodes: assumptions and premises. *Assumptions* are foundational data. Each environment in the label of a (non-assumption) datum node comprises a set of assumptions from which the datum can consistently be derived. *Premises* are similar to assumptions, but are taken to hold universally, and are not explicitly represented in environments. The task of the ATMS is to ensure that each node label is consistent, sound, complete and minimal. As the reasoner informs the ATMS of new datum nodes and justifications, the ATMS *label propagation* algorithms update the labels of previously asserted nodes to remove any subsumed environments (in the case of a normal justification), or any environments which subsume an environment (in the case of a new justification for the distinguished node $N_{\perp}$ which represents contradiction).

The ontology debugging problem can be mapped onto the operations of the ATMS in a straightforward way. Each TBOX axiom is represented by an ATMS assumption. For concreteness, we assume a TBOX $\Gamma = \{ax_1, \dots, ax_n\}$, where each axiom ax_i is of the form: $A_i \sqsubseteq C_i$ and all concept descriptions are in negation normal form (NNF). The assumption that each concept is non-empty, e.g., $A_i(c)$ for some constant c , is represented by an ATMS premise. The reasoner uses standard rules of inference to infer new concept instances from some consistent set of datum nodes (i.e., nodes whose labels do not subsume the label of $N_{\perp}$). A suitable list of rules that can be used by the reasoner to infer new justifications is shown in Figure 1. The process of creating the dependency graph terminates when no rule can be applied to any node of the graph. At this point, each environment of a node N_{datum} is a minimal set of axioms that can be used to derive $datum$, and the label of $N_{\perp}$ consists of sets of axioms responsible for clashes. In addition, the information given by justifications for nodes can be used to pinpoint the parts of axioms, e.g., concepts causing the clashes.

In conclusion, there is a clear mapping between the functionality provided by the ATMS and the problems of ontology debugging, and we believe that a systematic investigation of the practicality of using an ATMS for ontology debugging is a fruitful direction for future research.

REFERENCES

- [1] Johan de Kleer, ‘An assumption-based TMS’, *Artificial Intelligence*, **28**(2), 127–162, (March 1986).

⁴ With disjunctions, the setting is more complicated. However, disjunctions can be handled in several different ways, e.g., similarly to the extended ATMS described by de Kleer in [2], or by using sub-graphs to deal with the or-branches.

$\sqsubseteq$ -rule	If the current node is $N_{A_i(a)} : \langle A_i(a), L, J \rangle$ and $A_i \sqsubseteq C_i \in \Gamma$, then create a new node $N_{C_i(a)} : \langle C_i(a), L', \{(A_i(a))\} \rangle$ where $L' = \{e \cup \{ax_i\} e \in L\}$. If $N_{C_i(a)}$ exists, update its label and justifications. $L_{C_i(a)} = L_{C_i(a)} \cup L'$; $J_{C_i(a)} = J_{C_i(a)} \cup \{(A_i(a))\}$.
$\sqcap$ -rule	If the current node is $N_{C_i(a) \sqcap C_j(a)} : \langle C_i(a) \sqcap C_j(a), L, J \rangle$, then create 2 new nodes $N_{C_i(a)} : \langle C_i(a), L, \{(C_i(a) \sqcap C_j(a))\} \rangle$ $N_{C_j(a)} : \langle C_j(a), L, \{(C_i(a) \sqcap C_j(a))\} \rangle$ if they exist, update their labels and justifications.
$\exists$ -rule	If the current node is $N_{(\exists s.C)(a)} : \langle (\exists s.C)(a), L, J \rangle$, then create 2 nodes $N_{s(a,b)} : \langle s(a,b), L, \{(\exists s.C)(a)\} \rangle$ $N_{C(b)} : \langle C(b), L, \{(\exists s.C)(a)\} \rangle$ where b is a new individual.
$\forall$ -rule	If the current node is $N_{(\forall s.C)(a)} : \langle (\forall s.C)(a), L, J \rangle$ then if there exists $N_{s(a,b)}$, create a node $N_{C(b)} : \langle C(b), L, \{((\forall s.C)(a), s(a,b))\} \rangle$
$\perp$ -rule	If there exist 2 nodes $N_{A_i(a)} : \langle A_i(a), L_1, J_1 \rangle$ $N_{\neg A_i(a)} : \langle \neg A_i(a), L_2, J_2 \rangle$ in the graph, then create a new node $N_{\perp} : \langle \perp, \{e_1 \cup e_2 e_1 \in L_1, e_2 \in L_2\}, \{j_1 \cup j_2 j_1 \in J_1, j_2 \in J_2\} \rangle$. If $N_{\perp} : \langle \perp, L_{\perp}, J_{\perp} \rangle$ exists, update its label and justification: $L_{\perp} = L_{\perp} \cup \{e_1 \cup e_2 e_1 \in L_1, e_2 \in L_2\}$ $J_{\perp} = J_{\perp} \cup \{j_1 \cup j_2 j_1 \in J_1, j_2 \in J_2\}$

Figure 1. Rules for creating and updating nodes in the dependency graph

- [2] Johan de Kleer, ‘Extending the ATMS’, *Artificial Intelligence*, **28**(2), 163–196, (1986).
- [3] Jon Doyle, ‘A truth maintenance system’, *Artificial Intelligence*, **12**(3), 231–272, (1979).
- [4] Matthew Horridge, Bijan Parsia, and Ulrike Sattler, ‘Explaining inconsistencies in OWL Ontologies’, in *Proceedings of SUM '09*, pp. 124–137, Berlin, Heidelberg, (2009). Springer-Verlag.
- [5] A. Kalyanpur, B. Parsia, B.C. Grau, and E. Sirin, ‘Justifications for entailments in expressive description logics’, Technical report, University of Maryland, (2006).
- [6] A. Kalyanpur, B. Parsia, E. Sirin, and J. Hendler, ‘Debugging unsatisfiable classes in OWL ontologies’, *Journal of Web Semantics*, **3**(4), 268–293, (2005).
- [7] Joey Sik Chun Lam, Derek H. Sleeman, Jeff Z. Pan, and Wamberto Weber Vasconcelos, ‘A fine-grained approach to resolving unsatisfiable ontologies’, *J. Data Semantics*, **10**, 62–95, (2008).
- [8] Thomas Meyer, Kevin Lee, Richard Booth, and Jeff Z. Pan, ‘Finding maximally satisfiable terminologies for the description logic ALC’, in *Proceedings of AAAI 2006*. AAAI Press, (2006).
- [9] Stefan Schlobach and Ronald Cornet, ‘Non-standard reasoning services for the debugging of description logic terminologies’, in *Proceedings of IJCAI'03*, pp. 355–360. Morgan Kaufmann, (2003).
- [10] Stuart C. Shapiro, ‘Belief revision and truth maintenance systems: An overview and a proposal’, Technical report, SUNY-Buffalo, (1998).
- [11] Hai Wang, Matthew Horridge, Alan Rector, Nick Drummond, and Julian Seidenberg, ‘Debugging OWL-DL ontologies: A heuristic approach’, in *ISWC 2005, LNCS 3729*, pp. 745–757. Springer, (2005).

Qualitative Causal Analysis of Empirical Knowledge for Ontology Evolution in Physics

Jos Lehmann¹ and Alan Bundy² and Michael Chan³

1 INTRODUCTION

Ontology evolution and its automation are key factors for achieving software's flexibility and adaptability. In the approach to automated ontology evolution adopted in the GALILEO⁴ project, progress in physics is modelled as a process of ontology evolution. An overview of the approach is provided in Section 2. Section 3 shows that the construction and the modification of qualitative causal models of experimental set-ups make it possible to gain information about the quantities that appear in an equation and contribute to creating the logical conditions for the equation to evolve.

2 ONTOLOGY REPAIR PLANS

In the framework of the GALILEO project a number of so-called *Ontology Repair Plans* (ORPs) are being developed and implemented in higher-order logic [1]. ORPs detect and resolve a contradiction between two or more ontologies. In ORPs developed thus far, one of the ontologies represents a theory while the second ontology represents a sensory or experimental set-up for that theory. When the sensory ontology generates a theorem that contradicts a theorem of the theoretical ontology, an ORP is triggered which amends the two ontologies according to the observations. The development of ORPs is inspired by cases in the history of physics. So far, a few ORPs have been developed from a number of development cases, which reflect common strategies used in physics to cope with contradictory evidence. One of the ORPs is called *Where is my stuff?* (WMS) and was inspired by the discovery of latent heat.

Until the second half of the 18th century, the chemical/physical notion of heat was conflated with the notion of temperature and it was seen as a function of time – or of a temporal quantity, e.g. flow. Flow was defined as occurring when two physical bodies at different temperatures are in direct contact with one another. Equation 1 is a rational reconstruction⁵ of this pre-modern view:

$$Q = m \times \Delta t \quad (1)$$

where Q is the heat, measured by temperature, of a physical body, m is the mass of the body, Δt is the flow of heat, measured by time, from the hotter to the cooler object.

Around 1761 Joseph Black observed that (i) ice melts at constant temperature and (ii) the time required to melt a pound of ice is 140

times greater than the time required to raise a pound of water one degree in its temperature, both the ice and the water receiving the heat equally fast. This observation required to distinguish heat from temperature, thus ultimately change the very meaning of the quantity Q . Equation 1 evolved into:

$$Q = m \times \Delta T + m \times L \quad (2)$$

where Q is the heat put into or taken out of the body, m is the mass of the body, ΔT is the change in temperature, L is the specific latent heat required by a given substance during its phase transitions.

WMS's logical infrastructure emulates part of the evolution from Equation 1 to Equation 2 and adds to Equation 1 a component for the heat transferred during phase transitions. The equation for such intermediary theory would be: $Q = m \times \Delta t + Q_{\text{phase-transition}}$. Such addition-strategy is found in other cases in physics, e.g. the postulation of dark matter or of planets to account for unpredictable yet observed gravitational behaviour in galaxies or, resp., in planetary systems.

3 FROM THEORIES TO EXPERIMENTS

An aspect of the evolution of a physics theory that needs to be clarified is how the experimental set-up represented by the sensory ontology comes to produce evidence that contradicts the expectations of the theoretical ontology.

To this end a causal model of an experimental set-up for Equation 1 is discussed here. In particular, given the qualitative causal model shown in Figure 1, a new model is derived (Figure 2) based on principles 1 to 3 (see below). Running simulations on both models provides information (Figure 3) about the quantities that appear in the equation and create the conditions for the equation to evolve.

The causal models for Equation 1 are based on Qualitative Process Theory (QPT) [2], which allows to simulate the behaviour of a system through the explicit representation of causal relations between its quantities. The models and the results of the simulations were produced using a QTP-based tool called Garp3 (available on <http://hcs.science.uva.nl/QRM/>). In Garp3 terminology, a QPT model consists of a number of *model fragments* that describe their own sufficient or necessary conditions (in red resp. blue in the figures). Such fragments consist of *entity types* and *relations* between them, such as *Container*, *Substance* and *Contains*. Entities types have *quantities*, the value of which is a combination of their positive or negative *magnitude* on a qualitative scale⁶ and of a positive or neg-

¹ University of Edinburgh, United Kingdom, email: jlehmann@inf.ed.ac.uk

² University of Edinburgh, United Kingdom, email: bundy@inf.ed.ac.uk

³ University of Edinburgh, United Kingdom, email: mchan@inf.ed.ac.uk

⁴ Guided Analysis of Logical Inconsistencies Leads to Evolved Ontologies

⁵ At the time the theory of heat was not expressed in mathematical equations.

⁶ The qualitative values for *Temperature* are named after phases. A phase *Freeze_melt* is included between points *Frozen* and *Melted* to reflect the state of knowledge at the time of Equation 1: it was believed that temperature would change during phase transitions, which requires an interval between the solid and the liquid phases.

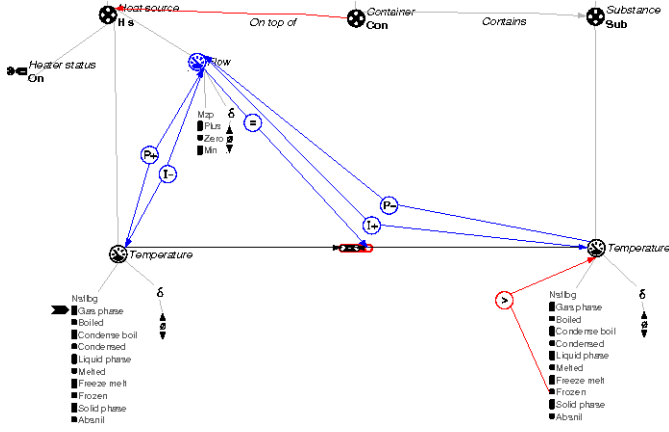


Figure 1. Process *Heat_flow*: model of experimental set-up for Equation 1

ative derivative (δ) that indicates whether the quantity is changing in magnitude. On the other hand, complex fragments represent dynamic features as a combination of causal relations and constraints on the quantities of the entities. Causal relationships come in two flavours:

- *influences*, $I^+(Cause, Effect)$ and $I^-(Cause, Effect)$, direct resp. inverse, indicate that the effect quantity changes if the cause quantity is non zero; examples in Figure 1 are $I^+(Flow(H_s), Temperature(Sub))$ and $I^-(Flow(H_s), Temperature(H_s))$.
- *proportionalities*, $P^+(Cause, Effect)$ and $P^-(Cause, Effect)$, direct resp. inverse, indicate that the effect quantity changes if the cause quantity changes; examples in Figure 1 are $P^+(Temperature(H_s), Flow(H_s))$ and $P^-(Temperature(Sub), Flow(H_s))$.

A *scenario* is a description of a state containing instances of the entities described in the model fragments. In Figure 3, for instance, H_2O , for which the temperature value histories are plotted, instantiates the variable *Sub* of type *Substance* of Figures 1 and 2; *Sto1* (for stove) instantiates *H_s* of type *Heat_source*. Given a QPT model of a system and a scenario, the qualitative simulation engine calculates the sequence of states that follow from the scenario, organising them in alternative sequences whenever the model contains ambiguities which allow for branching. One way of visualising a single sequence is through the value histories, as in Figure 3.

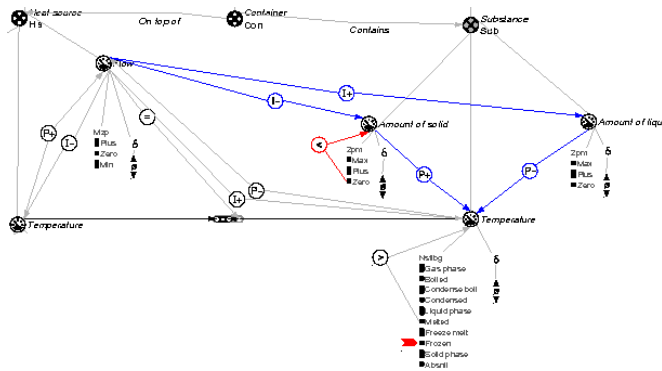


Figure 2. Process *Melting*: modified model of experimental set-up for Equation 1

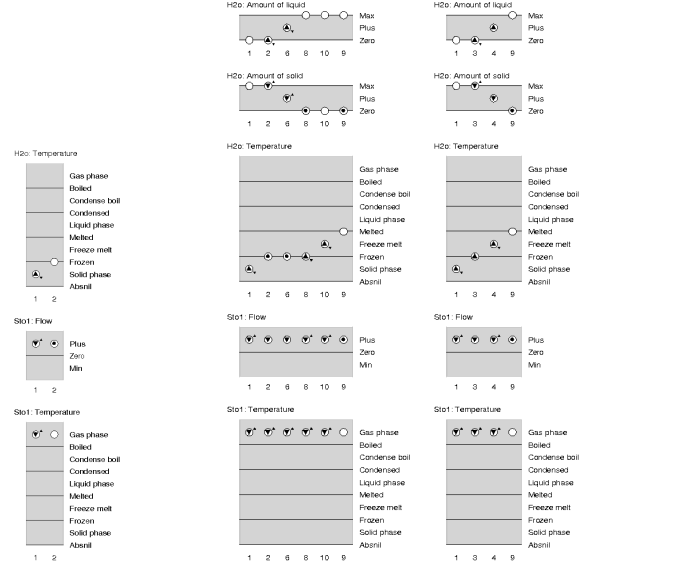


Figure 3. Left: value history for process *Heat_flow* consisting of 2 states. Right: two alternative value histories for process *Melting*, consisting of 6 resp. 4 states. The 6-states simulation envisions an interruption of the temperature rise, the 4-states matches the prediction of Equation 1

In order to test Equation 1 against phase changes the process *Heat_flow* shown in Figure 1 needs to be modified into the process *Melting*, which includes as a precondition *Heat_flow* (which is grayed-out Figure 2). *Melting* is activated for *Frozen* < *Temperature(Sub)* < *Melted* and includes quantities for phases (*Amount_of_solid* and *Amount_of_liquid*) in order to observe Equation 1 at work during phase changes. These are phenomenological quantities, not included in the original equation, and their causal role in the modified model *should be neutral from an energetic viewpoint*, their insertion in the model should be based on the following three principles:

1. changes in their values should be direct effects of the cause quantity (i.e. *Flow*);
2. they should indirectly affect the effect quantity (i.e. *Temperature*);
3. they should exert an opposite indirect causal influence on the cause quantity with respect to the influence exerted on it by the effect quantity.

These changes to the process *Heat_flow* create an ambiguity in the model for the quantity *Temperature*, which during the process *Melting* is at the same time positively directly influenced by *Flow* and indirectly negatively proportional to it. Such ambiguity is reflected in the two alternative simulations produced by the modified model (Figure 3). The first simulation envisions an interruption of the temperature rise, whereas the second matches the prediction based on Equation 1. The very creation of the ambiguity through the modification steps 1 to 3 above sheds light on how the contradiction between the theoretical and the sensory ontology is generated.

REFERENCES

- [1] A. Bundy and M. Chan, 'Towards ontology evolution in physics', W. Hodges, editor, *Procs. Wollic 2008. LNCS Springer-Verlag*, (2008).
- [2] K.D. Forbus, 'Qualitative process theory', *Artificial Intelligence, Vol 24*, 85-168, (1984).

Uncertainty Reasoning through Similarity in Context

Claudia d'Amato and Nicola Fanizzi¹

1 SIMILARITY, INDUCTION & CONTEXTS

Purely deductive and logic-based methods for semantic knowledge bases (KBs) expressed in *Description Logics* (DLs) may fall short due to the heterogeneity of the data-sources and the inherent incompleteness. Inductive methods may be better suited, as they are often both efficient and noise-tolerant. In particular, methods grounded in similarity have been proposed to provide viable and robust methods for many difficult tasks, casting them as inductive problems [3, 7].

Many inductive methods for ontologies expressed in DL-based languages rely on a notion of similarity or distance [13]. Most of the measures in the literature focus on *concept-similarity* [1]. However, instance-based reasoning techniques borrowed from *Machine Learning* require a notion of similarity between *individuals*. Some measures for specific DLs have been proposed [2, 4] they are still partly based on structural criteria (a notion of normal form) which determine their main weakness: language-dependence. To overcome these limitations, language-independent measures for individuals based on their *behavior* w.r.t. a committee of features (expressed as DL concepts), say $C = \{F_1, F_2, \dots, F_m\}$, which acts as *context of reference* [10, 11]. The rationale was that a context stands as a group of discriminating *features*.

Formally, a family of semantic metrics for individuals, inspired to Minkowski's norms, was defined as follows [6, 3]:

Given a context $C = \{F_1, F_2, \dots, F_m\}$ for the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a family of dissimilarity functions for individuals occurring in the ABox $\mathcal{A}$ ($\text{Ind}(\mathcal{A})$) is defined $d_p^C: \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mapsto [0, 1]$ with $\forall a, b \in \text{Ind}(\mathcal{A}): d_p^C(a, b) = \sqrt[p]{\sum_{i=1}^m |\delta_i(a, b)|^p}$, where $p > 0$ and the i^{th} -dissimilarity function δ_i is defined by:

$$\delta_i(a, b) = \begin{cases} 0 & [\mathcal{K} \models F_i(a) \wedge \mathcal{K} \models F_i(b)] \vee [\mathcal{K} \models \neg F_i(a) \wedge \mathcal{K} \models \neg F_i(b)] \\ 1 & [\mathcal{K} \models F_i(a) \wedge \mathcal{K} \models \neg F_i(b)] \vee [\mathcal{K} \models \neg F_i(a) \wedge \mathcal{K} \models F_i(b)] \\ u_i & \text{otherwise} \end{cases}$$

Here u_i stands for the uncertainty related to the i -th feature. In lack of prior knowledge, the features are assumed to have uniform priors.

2 LEARNING CONTEXTS

The choice of suitable context is important for the similarity assessment. Depending on their number, one may adopt (a subset of) the concepts defined in the KB or features can be generated *ad hoc* in advance. Indeed, some features may likely have a higher *discriminating power* w.r.t. the others, then they should turn out to be more relevant in determining the similarity of the individuals in a given context. Moving from this observation, an extension of the measures proposed in [6] is presented, where each feature of the committee may be weighted on the grounds of the amount of information it conveys [3].

In [6, 7], we propose feature selection methods based on stochastic search which can be employed to determine a context that is able to properly discriminate the individual resources. Encoding a given criterion in a fitness function, the methods perform a randomized search through customized versions of well known algorithms (e.g. simulated annealing or genetic programming).

Even more so, once the context of features has been produced, each concept may give a different contribution to the similarity. That is why normally the values of the dissimilarity measures in the metrics δ_i are generally weighted: the more they can split the given instances in an original way the higher this weight (then they are also normalized). The ability of splitting the individuals has been measured in terms of the related *entropy* or by recurring to a measure of their *sample variance*.

3 APPLICATIONS

We present two applications: one concerns the usage of contextual similarity to perform inductive classification; the second allows for the definition and reasoning with vague concepts.

3.1 A Distance-based Classification Method

Individual classification or search through a k -NN method [13] has been customized for DLs [2, 3]. The method is ascribed to the category of lazy instance-based learning, since the mere learning phase is reduced to memorizing examples and counter-examples of the target concept. Then, during the classification phase, a new instance is classified in analogy with its neighbors.

More formally, the method aims at inducing an approximation for a hypothesis function h ranging in a set of values $V = \{v_1, \dots, v_s\}$ standing for the (disjoint) cases to be predicted. Normally a limited number of training instances is needed for the classification, especially if they are prototypical for a region of the instance space. Let x_q be the query instance whose membership w.r.t. some query concept is to be determined. Using a metric, the set of the k nearest training instances w.r.t. x_q is selected: $NN(x_q) = \{x_i \mid i = 1, \dots, k\}$. Then the k -NN procedure approximates h for classifying x_q on the grounds of the value that h assumes for the instances in $NN(x_q)$. The estimated value for $h(x_q)$ is decided through a *weighted majority vote*, w.r.t. the similarity of the neighbor individual. Formally:

$$\hat{h}(x_q) = \underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^k \mathbb{I}(v, h(x_i)) / d(x_i, x_q)^b$$

where the function $\mathbb{I}$ returns 1 in case of matching arguments and 0 otherwise. We adopt the set $V = \{-1, 0, +1\}$, where $+1$ stands for membership, -1 for non-membership, and 0 for uncertainty.

Note that, being this procedure based on a majority vote of the individuals in the neighborhood, it is less error-prone in case of noise

¹ Università di Bari, "A. Moro", Italy – contact: fanizzi@di.uniba.it

in the data (e.g. incorrect assertions) w.r.t. a purely logic deductive procedure. Therefore it may be able to give a correct classification even in case of inconsistent assertions. Obviously the procedure can be forced to give a positive/negative answer providing training individuals labeled with values ± 1 only.

Similar approaches have been followed for deriving semantic kernels encoding aspects of contextual similarity used for non-parametric statistical learning [9].

3.2 Representing Vague Concepts in Rough DL

Modeling vague concepts has often been tackled through numeric approaches. A drawback of these approaches is that uncertainty is introduced in the model, which often has the consequence that crisp answers to queries cannot be returned. *Rough DLs* [12] introduced a mechanism that allows for modeling vague knowledge by means of a crisp specification of approximations of a concept. Two crisp concepts can approximate an underspecified concept as sub-/super-concepts, describing which elements are definitely/possibly, instances of the vague concept.

Rough DLs extend classical DLs with two modal-like operators. The approximations are based on an indiscernibility relation R among individuals. The *upper approximation* of a concept includes individuals that are indiscernible from at least another one that is known to belong to the concept: $\overline{C} = \{a \mid \exists b : R(a, b) \wedge b \in C\}$. Similarly, the *lower approximation* is $\underline{C} = \{a \mid \forall b : R(a, b) \rightarrow b \in C\}$. Intuitively, $\overline{C}$ of a concept C includes elements with the *typical* properties of C , whereas $\underline{C}$ contains the *prototypical* instances of C . Given the proposed metrics for individuals, they may be employed to induce the definition of equivalence relations to be embedded into Rough DL KBs [8]. Therefore, standard reasoners could be used.

Let N_I the set of individual names and $F_i \in \mathbf{C}$ be a feature concept. The *projection* $\pi_i : N_I \mapsto \{0, \frac{1}{2}, 1\}$, given an individual $a \in N_I$ assigns to $\pi_i(a)$ the values 1 if $\mathcal{K} \models F_i(a)$, 0 if $\mathcal{K} \models \neg F_i(a)$ and $\frac{1}{2}$ otherwise. Two individuals, say a and b , are indiscernible w.r.t. the context $\mathbf{C}$ iff $\forall i \in \{1, \dots, m\} : \pi_i(a) = \pi_i(b)$. Then, the *indiscernibility relation* R_C induced by $\mathbf{C}$ is defined as: $R_C = \{(a, b) \in N_I \times N_I \mid \forall i \in \{1, \dots, m\} : \pi_i(a) = \pi_i(b)\}$.

Any indiscernibility relation partitions N_I (and then the domain of an interpretation) into sets, i.e. equivalence classes (*elementary sets*) denoted, for a generic individual a , with $[a]_C$. In turn, each class induces a concept, denoted C_a . The extension of a *C-definable* concept corresponds to one of the union of elementary sets. The other concepts may only be approximated. Let D be a generic DL concept description. The *contextual upper and lower approximations* of D w.r.t. $\mathbf{C}$, denoted respectively with $\overline{D}^C$ and $\underline{D}^C$, are defined: $\overline{D}^C = \{a \in N_I \mid C_a \sqcap D \not\models \perp\}$ and $\underline{D}^C = \{a \in N_I \mid C_a \sqsubseteq D\}$.

A less strict type of approximation is introduced, based on the notion of *tolerance* [5]. Exploiting the dissimilarity functions, it is easy to extend this kind of (contextual) approximation to the case of Rough DL. Let a *tolerance function* on a set U be any function $\tau : U \times U \mapsto [0, 1]$ such that $\forall a, b \in U, \tau(a, b) = 1$ and $\tau(a, b) = \tau(b, a)$. Given a tolerance function τ on U and a *threshold* $\theta \in [0, 1]$, a *neighborhood function* $\nu : U \mapsto 2^U$ is defined $\nu_\theta(a) = \{b \in U \mid \tau(a, b) \geq \theta\}$. For each $a \in U$, the set $\nu_\theta(a)$ is also called the neighborhood of a .

Now, consider $N_I = U$ as universal set and a dissimilarity function d_p^C (for some context $\mathbf{C}$) to define a tolerance function and a threshold $\theta \in [0, 1]$. It is easy to derive an equivalence relationship on N_I , where the classes are made up of individuals within a certain degree of similarity, indicated by the threshold: $[a]_C = \nu_\theta(a)$. The

notions of upper and lower approximation w.r.t. the induced equivalence relationship descend straightforwardly. These approximations depend on the threshold, then there is a way to control the degree of indiscernibility needed to model vague concepts. This applies also in the standard Rough DL setting.

4 OUTLOOK

We explored here two possible application of (dis)similarity measures in a given context represented by a set of concepts within DL KBs. As mentioned, further developments may come by the definition of kernel functions which can be easily plugged into kernel machines to perform a wealthy of tasks such as contextual retrieval, ranking, outlier and novelty detection, clustering, and so on.

REFERENCES

- [1] A. Borgida, T. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In I. Horrocks et al., editors, *Working Notes of the International Description Logics Workshop*, volume 147 of *CEUR Workshop Proceedings*, 2005.
- [2] C. d'Amato, N. Fanizzi, and F. Esposito. Analogical reasoning in description logics. In P. da Costa et al., editors, *Uncertainty Reasoning for the Semantic Web I*, volume 5327 of *LNAI*, pages 336–354. Springer, 2008.
- [3] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: An inductive approach. In S. Bechhofer et al., editors, *Proceedings of the 5th European Semantic Web Conference, ESWC2008*, volume 5021 of *LNCS*, pages 288–302. Springer, 2008.
- [4] C. d'Amato, S. Staab, and N. Fanizzi. On the influence of Description Logics ontologies on conceptual similarity. In A. Gangemi and J. Euzenat, editors, *Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management, EKAW2008*, volume 5268 of *LNCS*, pages 48–63. Springer, 2008.
- [5] P. Doherty, M. Grabowski, W. Lukasiewicz, and A. Szalas. Towards a framework for approximate ontologies. *Fundamenta Informaticae*, 57(2–4):147–165, 2003.
- [6] N. Fanizzi, C. d'Amato, and F. Esposito. Induction of optimal semi-distances for individuals based on feature sets. In D. Calvanese et al., editors, *Working Notes of the 20th International Description Logics Workshop, DL2007*, volume 250 of *CEUR Workshop Proceedings*, 2007.
- [7] N. Fanizzi, C. d'Amato, and F. Esposito. Metric-based stochastic conceptual clustering for ontologies. *Information Systems*, 34(8):725–739, 2009.
- [8] N. Fanizzi, C. d'Amato, F. Esposito, and T. Lukasiewicz. Representing uncertain concepts in rough description logics via contextual indiscernibility relations. In F. Bobillo et al., editors, *Proceedings of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web, URSW2008*, volume 423 of *CEUR Workshop Proceedings*, 2008.
- [9] N. Fanizzi, C. d'Amato, and F. Esposito. Statistical learning for inductive query answering on OWL ontologies. In A. Sheth et al., editors, *Proceedings of the 7th International Semantic Web Conference, ISWC2008*, volume 5318 of *LNCS*, pages 195–212. Springer, 2008.
- [10] R. Goldstone, D. Medin, and J. Halberstadt. Similarity in context. *Memory and Cognition*, 25(2):237–255, 1997.
- [11] K. Janowicz. Kinds of contexts and their impact on semantic similarity measurement. In *Proceedings of the 6th IEEE International Conference on Pervasive Computing and Communications, PerCom2008*, pages 441–446. IEEE, 2008.
- [12] S. Schlobach, M. C. A. Klein, and L. Peelen. Description logics with approximate definitions - precise modeling of vague concepts. In M. Veloso, editor, *Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 2007*, pages 557–562, Hyderabad, India, 2007.
- [13] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search – The Metric Space Approach*. Advances in Database Systems. Springer, 2007.

Ontology Archaeology: Mining a Decade of Effort on the Suggested Upper Merged Ontology

Adam Pease, Chris Benzmueller¹

1 EXTENDED ABSTRACT

The Suggested Upper Merged Ontology [5] is a large ontology defined in first-order logic with some higher-order extensions [1]. The project began in the year 2000. Each version has been released open source and publicly from the start, which provides a unique record of the construction of a formal ontology. While initially just an upper ontology, it now encompasses a wide variety of domains, and some recent work has involved semi-automatically merging large factbases with the fully axiomatized hand-built content [3]. SUMO has been mapped by hand to all of English WordNet [6] and several other languages (Elkateb et al 2006; Borra et al 2010; Pease& Fellbaum, 2010) and is used in natural language understanding tasks (Pease & Li, 2010). SUMO is supported by tools for ontology development (Pease, 2003) and inference (Trac et al 2007) and used in the yearly CASC theorem proving competition (Pease et al 2010).

Before discussing some of the specifics of the development history, we must note that the historical data is still incomplete. A change in employment of the first author resulted in some data on the development of the domain ontologies being lost prior to mid-2004, when Articulate Software was founded. This accounts for the large non-linearity in the graph shown in Figure 1 at that time.

For the first two years, development was confined to the original upper level effort: the Suggested Upper Merged Ontology. The first step in the project involved seeding the ontology with the contents of all general-purpose, and formally defined axioms that we could find in open, academic work and merging those theories into a common structure. This resulted in about 5700 lines of SUO-KIF code that was released May 11, 2001. After that point, the majority of axioms were developed by the SUMO project team, although many more small theories and additions were provided by others over the entire history. All contributions are credited in the CVS logs for the ontologies [13]

In June 2002 a large US government project provided support for creating a host of new domain ontologies that would extend SUMO. These ontologies covered information about economy, finance, government, geography and many other topics necessary for encoding facts about the world's geo-political situation. It is these domain ontologies which would form the largest addition of domain content to SUMO for some time, and which is responsible in the graph for the first major jump in content size.

Early in 2003 we began a comprehensive project to add content that would structure and define concepts more general than the new domain ontologies, yet more specific than SUMO

itself. This would become known as the Mid-Level Ontology (MILO). The methodology for creating MILO was an outgrowth of our work in mapping SUMO to WordNet (Niles&Pease, 2003). After the original mapping was complete, it was clear that a vast number of linguistic terms (or what in WordNet are called synsets) lacked a mapping to an equivalent formally defined concept in SUMO. We therefore set out to create roughly equivalent formal concepts for every synset that appeared at least 3 times in the WordNet semantic concordance, or SemCor (Miller et al 1993). Since the completion of that effort roughly a year later, MILO has continued to be an active site for development of new content as additional domains are covered, and new intermediate-level content is required to structure and adequately define terms that apply to many domains. More recent significant new domains that can be seen increasing the size of the overall theory were many military concepts added in 2006 and digital media concepts in 2010.

While SUMO has always viewed manual creation of formal axioms as the primary effort, the breadth and formality of content has recently enabled databases of lightly structured information to be integrated automatically, and to take advantage of the definitions that SUMO provides. Simple factbases alone provide little opportunities for significant inferences, but combined with thousands of formal rules, many new and useful conclusions become possible through automated inference. The mappings to Mondial, portions of DBPedia, the Open Biomedical Ontologies, and YAGO have been significant. The content of these databases dwarfs the hand-created content in SUMO, as shown in Figure 2, which shows the same time period as Figure 1, but with, in order from the bottom, DBPedia, OBO and YAGO added.

SUMO			
Total Terms	relations	Total Axioms	Rules
1173	353	4741	932
MILO			
Total Terms	relations	Total Axioms	Rules
1662	159	5116	1183
Domain ontologies			
Total Terms	relations	Total Axioms	Rules
17312	708	77974	2041
Total for all ontologies			
Total Terms	relations	Total Axioms	Rules
20147	1220	87831	4156

Table 1: SUMO term and axiom statistics

¹ Articulate Software, Angwin, CA, USA. Email: {apease, cbenzmueller}@articulatesoftware.com.

Lines of code are a useful approximation of the progress of ontology development, but insufficient. The Sigma browser provides statistics of how many terms are in the knowledge base, and how many of those terms are relations. It also shows how many axioms there are, and how many of those axioms are "if..then" rules. Those numbers are shown in Table 1. Note that we do not include totals for YAGO, which dwarfs the hand-coded content, and totals some 16,425,285 axioms.

REFERENCES

- [1] Benzmueller, C., and Pease, A., (2010). Progress in Automating Higher-Order Ontology Reasoning, to appear in Practical Aspects of Automated Reasoning workshop of CADE 2010..
- [2] Borra, A., Pease, A., Roxas, R., Dita, S., (2010). Introducing Filipino WordNet, in Principles, Construction and Application of Multilingual Wordnets: Proc. of the 5th Global WordNet Conf., Bhattacharyya, et al, editors, ISBN:978-81-8487-083-1.
- [3] de Melo, G., Suchanek, F., and Pease, A., (2008). Integrating YAGO into the Suggested Upper Merged Ontology. In Proceedings of the 20th IEEE Int'l Conf. on Tools with AI (ICTAI 2008).
- [4] Elkateb, S., Black, W., Rodriguez, H., Alkhalifa, M., Vossen, P., Pease, A. and Fellbaum, C., (2006). Building a WordNet for Arabic, in Proceedings of LREC 2006.
- [5] Miller, G. A., Leacock, C., Teng, R., and Bunker, R. T. 1993. A semantic concordance. In Proceedings of the Workshop on Human Language Technology (Princeton, New Jersey, March 21 - 24, 1993). Morristown, NJ, 303-308.
- [6] Niles, I., & Pease, A., (2001), Toward a Standard Upper Ontology, in Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), C. Welty and B. Smith, eds..
- [7] Niles, I., and Pease, A., (2003). Linking Lexicons and Ontologies: Mapping WordNet to the Suggested Upper Merged Ontology, Proc. of the IEEE Int'l Conf. on Information and Know. Eng. pp 412-416.
- [8] Pease, A., (2003). The Sigma Ontology Development Environment, in Working Notes of the IJCAI-2003 Workshop on Ontology and Distributed Systems. Volume 71 of CEUR Workshop Proc. series.
- [9] Pease, A., and Fellbaum, C., (2010) Formal Ontology as Interlingua: The SUMO and WordNet Linking Project and GlobalWordNet, In: Huang, C. R. et al (eds.) Ontologies and Lexical Resources. Cambridge: Cambridge University Press, ISBN-13: 9780521886598.
- [10] Pease, A., and Li, J. (2010) Controlled English to Logic Translation. In Theory and Applications of Ontology, ed. Roberto Poli, Michael Healy, and Achilles Kameas, Springer, ISBN: 978-90-481-8846-8.
- [11] Pease, A., Sutcliffe, G., Siegel, N., and Trac, S., (2010). Large Theory Reasoning with SUMO at CASC, AI Communications, Volume 23, Number 2-3 / 2010, Special issue on Practical Aspects of Automated Reasoning, IOS Press, ISSN 0921-7126, pp 137-144.
- [12] Trac, S., Sutcliffe, G., and Pease, A., (2008) Integration of the TPTPWorld into SigmaKEE. Proceedings of IJCAR '08 Workshop on Practical Aspects of Automated Reasoning (PAAR-2008). Volume 373 of the CEUR Workshop Proceedings.
- [13] CVS logs for SUMO and its domain ontologies: <http://www.ontologyportal.org/SUMOhistory/>, <http://sigmakee.cvs.sourceforge.net/sigmakee/KBs/>

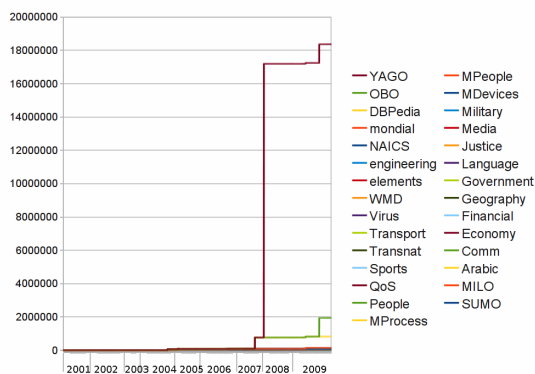


Figure 2: Automatic additions of database content December 2000 to present

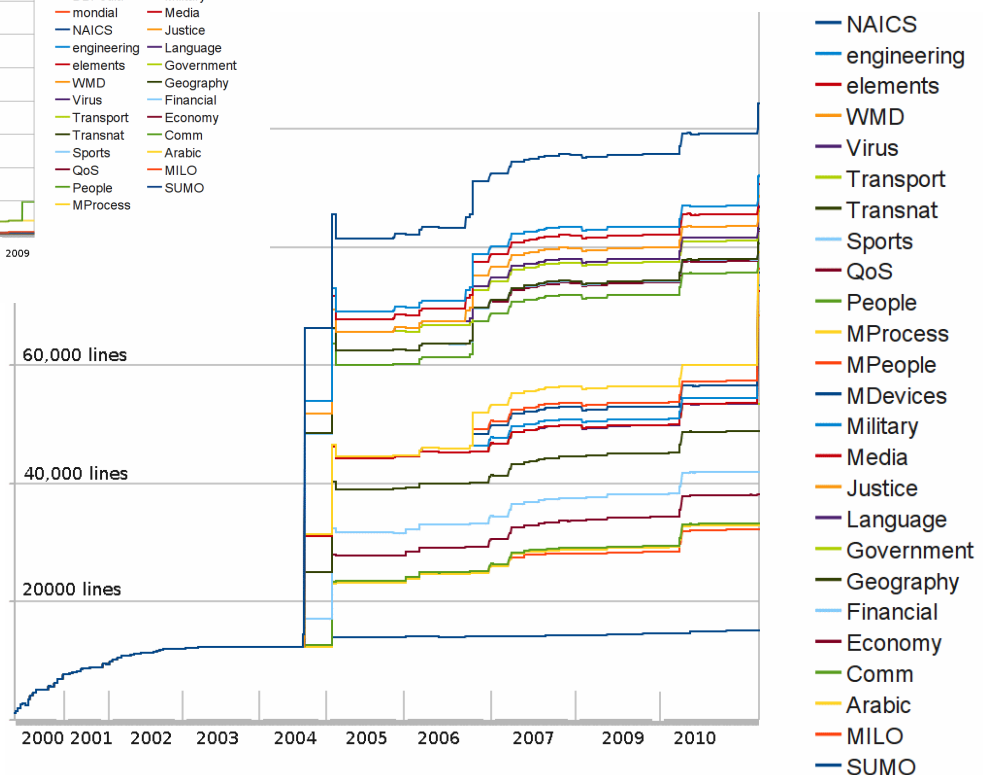


Figure 1: SUMO Family of Ontologies: Lines of Code Totals from December 2000 to Present

First Steps in the Computation of Root Justifications

Thomas Meyer and Kodylan Moodley¹ and Ivan Varzinczak²

1 INTRODUCTION

Description Logics (DLs) are widely accepted as an appropriate class of knowledge representation languages to represent and reason about ontologies [2]. Tools for performing standard reasoning tasks such as satisfiability and consequence checking have grown increasingly powerful and sophisticated in the last decade [3, 7]. In section 2 we review the standard approach for the resolution of modelling errors encountered in ontologies and propose the first steps in a new method for resolving such errors in section 3. The method is based on the notion of *root justifications* which we define and discuss. The approach we describe is applicable to a wide class of DLs. We don't provide a comprehensive formal introduction to DLs, but rather point the reader to the book by Baader et al. [2]. For our purposes a DL *TBox* consists of a finite set of *axioms* specifying the terminological part of an ontology. The Tbox includes (but need not be limited to) *subsumption statements* of the form $E \sqsubseteq F$ where E and F are (possibly complex) concept descriptions, built up from basic concepts. The semantics of DLs is based on the classical model theory for first-order logic. A DL interpretation I contains a non-empty *domain* Δ^I of elements and a mapping which interprets a basic concept A as a subset A^I of Δ^I . For purposes of illustration we shall assume that complex concepts can be constructed using negation ($\neg E$) and conjunction ($E \sqcap F$), where $\neg E$ is interpreted as $\Delta^I \setminus E^I$, and $E \sqcap F$ is interpreted as $E^I \cap F^I$. However, the inclusion of negation and conjunction is not a requirement. In addition, there may be other ways of constructing complex concepts.

An interpretation I is a *model* of a Tbox axiom $E \sqsubseteq F$ if and only if $E^I \subseteq F^I$. Given a Tbox Γ , a subsumption statement $E \sqsubseteq F$, and a basic concept A , (i) Γ is *A-unsatisfiable* if and only if for all models I of Γ , $A^I = \emptyset$, and (ii) $E \sqsubseteq F$ is a *consequence* of Γ if and only if every model of all axioms in Γ is also a model of $E \sqsubseteq F$.

2 ONTOLOGY DEBUGGING AND REPAIR

The *A-unsatisfiability* of Γ may be an indication that Γ contains a modelling error. Concept unsatisfiability is a special case of identifying *unwanted axioms* to eliminate modelling errors in ontologies. In general, ontology construction is an iterative process. During each iteration a potential ontology is constructed, a domain expert identifies *unwanted consequences* of the ontology, and (minimal) modifications are made to the ontology to ensure that unwanted consequences are eliminated. Formally, we are provided with a Tbox Γ and an unwanted axiom U with the requirement that $\Gamma \not\models U$.

In order to eliminate an unwanted axiom it is useful to determine the possible *causes* of the axiom being a consequence of Γ . A subset J of Γ is a *U-justification* for Γ if and only if $J \models U$ and for

every $J' \subset J$, $J' \not\models U$ [1]. The notion of a *U-justification* is a generalisation of a minimal unsatisfiability preserving sub-Tbox (MUPS) [6], where the latter applies to unsatisfiable concepts. We denote by $\mathcal{J}_\Gamma(U)$ the set of all *U-justifications* for Γ . As a running example in this paper, consider the following four Tbox axioms:

$$1. C \sqsubseteq A \quad 2. C \sqsubseteq \neg A \quad 3. F \sqsubseteq C \sqcap \neg A \quad 4. F \sqsubseteq C$$

Let Γ be the set containing the four axioms above. We represent Γ as the set $\{1, 2, 3, 4\}$ with the understanding that the natural numbers contained in the set are indices, representing the four axioms. Using the same notation, and taking the axiom $F \sqsubseteq \perp$ to be an unwanted axiom, it can be verified that $\mathcal{J}_\Gamma(F \sqsubseteq \perp) = \{\{1, 3\}, \{1, 2, 4\}\}$.

Justifications are useful for a number of reasons. They allow for the *pinpointing* of the causes of modelling errors. In our example they show that axioms 1 and 3 may not both occur in the ontology without having $F \sqsubseteq \perp$ as a consequence. Similarly axioms 1, 2, and 4 may not all occur in the ontology without having $F \sqsubseteq \perp$ as a consequence. In practice it is frequently the case that justifications are significantly smaller than the Tbox as a whole.

Justifications can also be used to perform *ontology repair*. A subset R of a Tbox Γ is a *U-repair* for Γ if and only if $R \not\models U$, and for every R' such that $R \subset R' \subseteq \Gamma$, $R' \models U$. We denote by $\mathcal{R}_\Gamma(U)$ the set of *U-repairs* for Γ . For our example it can be verified that $\mathcal{R}_\Gamma(F \sqsubseteq \perp) = \{\{2, 3, 4\}, \{1, 4\}, \{1, 2\}\}$.

A subset D of Γ is a *U-diagnosis* for Γ if and only if $D \cap J \neq \emptyset$ for every $J \in \mathcal{J}_\Gamma(U)$. D is a *minimal U-diagnosis* for Γ if and only if there is no *U-diagnosis* D' for Γ such that $D' \subset D$. The set of minimal *U-diagnoses* for Γ , denoted by $\mathcal{D}_\Gamma(U)$, can be used to generate all the *U-repairs* for Γ as follows [5, 1, 6]:

Theorem 1 $\mathcal{R}_\Gamma(U) = \{\Gamma \setminus D \mid D \in \mathcal{D}_\Gamma(U)\}$.

For our example $\mathcal{D}_\Gamma(F \sqsubseteq \perp) = \{\{1\}, \{2, 3\}, \{3, 4\}\}$ from which it can be verified, as we have seen, that $\mathcal{R}_\Gamma(F \sqsubseteq \perp) = \{\{2, 3, 4\}, \{1, 4\}, \{1, 2\}\}$. There are efficient methods for generating *U-repairs* from the *U-justifications*, with Reiter's *hitting set algorithm* [5], and variants of it, probably being the best known.

So far we have dealt with a single unwanted axiom, but as the discussion above indicates, it may well be that a domain expert identifies a *set* $\mathcal{U}$ of unwanted consequences. We are interested in (i) finding the causes of the unwanted axioms, and (ii) repairing the Tbox Γ by replacing it with a Tbox Γ' with the requirement that $\Gamma' \not\models U$ for every $U \in \mathcal{U}$. This is a generalisation of the idea of finding minimal incoherence-preserving sub-TBoxes (MIPS) as a way of eliminating all unsatisfiable concepts in a Tbox [6]. Finding the causes of the unwanted axioms is a matter of generating *all U-justifications* for every $U \in \mathcal{U}$. We denote the set of all such *U-justifications* by $\mathcal{J}_\Gamma(\mathcal{U})$. That is, $\mathcal{J}_\Gamma(\mathcal{U}) = \bigcup_{U \in \mathcal{U}} \mathcal{J}_\Gamma(U)$.

For our example, let $\mathcal{U} = \{F \sqsubseteq \perp, C \sqsubseteq \perp\}$. We have already seen that $\mathcal{J}_\Gamma(F \sqsubseteq \perp) = \{\{1, 3\}, \{1, 2, 4\}\}$. It is easily seen that $\mathcal{J}_\Gamma(C \sqsubseteq \perp) = \{\{1, 2\}\}$ and therefore that $\mathcal{J}_\Gamma(\mathcal{U}) =$

¹ Meraka Institute, CSIR and University of Kwazulu-Natal, South Africa. {tmeyer, kmoodley}@csir.co.za

² Meraka Institute, CSIR, South Africa. ivarzinczak@csir.co.za

$\{\{1, 3\}, \{1, 2, 4\}, \{1, 2\}\}$.

For Tbox repair our goal is to find the $\mathcal{U}$ -repairs for Γ . A subset R of Γ is a $\mathcal{U}$ -repair for Γ if and only if $R \not\models U$ for every $U \in \mathcal{U}$, and for every R' for which $R \subset R' \subseteq \Gamma$, $R' \models U$ for some $U \in \mathcal{U}$. We denote the set of $\mathcal{U}$ -repairs for Γ by $\mathcal{R}_\Gamma(\mathcal{U})$. For our example it can be verified that $\mathcal{R}_\Gamma(\mathcal{U}) = \{\{2, 3, 4\}, \{1, 4\}\}$.

An obvious method for computing Tbox repair is to eliminate unwanted axioms sequentially using existing methods for repair appropriate for dealing with a single unwanted axiom. However, the naïve approach to do so is not guaranteed to generate *only* $\mathcal{U}$ -repairs (i.e., elements of $\mathcal{R}_\Gamma(\mathcal{U})$): Suppose, in our example, that we decide to eliminate $F \sqsubseteq \perp$ first (followed by the elimination of $C \sqsubseteq \perp$). As we have seen on the previous page, the $(F \sqsubseteq \perp)$ -repairs for Γ are $\Gamma_1 = \{2, 3, 4\}$, $\Gamma_2 = \{1, 4\}$, and $\Gamma_3 = \{1, 2\}$. Having eliminated $F \sqsubseteq \perp$, we then move on to eliminating $C \sqsubseteq \perp$ from each Γ_i , for $i = 1, 2, 3$. It is easy to see that $\Gamma_1 \not\models C \sqsubseteq \perp$ and $\Gamma_2 \not\models C \sqsubseteq \perp$, but that $\Gamma_3 \models C \sqsubseteq \perp$. We therefore leave Γ_1 and Γ_2 unchanged, but we need to obtain the $(C \sqsubseteq \perp)$ -repairs for Γ_3 . It can be verified that the $(C \sqsubseteq \perp)$ -repairs for Γ_3 are $\{\{2\}\}$ and $\{\{1\}\}$. We thus have, as candidates for the $\mathcal{U}$ -repairs of Γ , the sets $\Gamma_1 = \{2, 3, 4\}$ and $\Gamma_2 = \{1, 4\}$, as well as the two $(C \sqsubseteq \perp)$ -repairs of Γ_3 : $\{2\}$ and $\{1\}$. But observe that the two $(C \sqsubseteq \perp)$ -repairs of Γ_3 are not $\mathcal{U}$ -repairs for Γ (we will refer to such sets as *false* repairs).

It can be shown that the process described above (i) will generate subsets of $\mathcal{U}$ -repairs for Γ only, and (ii) will generate at least all $\mathcal{U}$ -repairs for Γ . From this it follows that false $\mathcal{U}$ -repairs can be identified and removed: they will all be strict subsets of the $\mathcal{U}$ -repairs for Γ . Nevertheless, it would be useful, for the sake of efficiency, to eliminate the *generation* of such false $\mathcal{U}$ -repairs altogether.

It is possible to do better than the naïve approach described above by making an *informed* choice about which unwanted axioms to eliminate first. Suppose that, in our example, and in contrast to what we did above, we choose to eliminate $C \sqsubseteq \perp$ first (followed by the elimination of $F \sqsubseteq \perp$). It can be verified that one of the $(C \sqsubseteq \perp)$ -repairs for Γ is the set $\{2, 3, 4\}$, which also turns out to be a $(F \sqsubseteq \perp)$ -repair for Γ . The reason for it being a $(F \sqsubseteq \perp)$ -repair for Γ as well, is that one of the $(C \sqsubseteq \perp)$ -justifications for Γ ($\{1, 2\}$) is a strict subset of one of the $(F \sqsubseteq \perp)$ -justifications for Γ ($\{1, 2, 4\}$). In this case it will thus be more efficient to choose $C \sqsubseteq \perp$ as the unwanted axiom to be eliminated first, since we get the elimination of $F \sqsubseteq \perp$ for free.

This heuristic can be formalised by drawing a distinction between *root* and *derived* unwanted axioms [4]. Formally, an unwanted axiom U is a *derived* unwanted axiom for Γ if and only if there exists a $\mathcal{U}$ -justification J for Γ and a $\mathcal{U}'$ -justification J' for Γ such that $J' \subset J$. U is a *root* unwanted axiom for Γ if and only if it is *not* a derived unwanted axiom for Γ . The goal is to eliminate root unwanted axioms first with the expectation that in the process of doing so, other unwanted axioms may be eliminated as well. In our example $F \sqsubseteq \perp$ is a derived unwanted axiom for Γ since there is a $(F \sqsubseteq \perp)$ -justification for Γ ($\{1, 2, 4\}$) which is a strict superset of a $(C \sqsubseteq \perp)$ -justification for Γ ($\{1, 2\}$), while $C \sqsubseteq \perp$ is a root unwanted axiom for Γ . According to this heuristic we should therefore choose to eliminate the unwanted axiom $C \sqsubseteq \perp$ first.

Unfortunately the use of root unwanted axioms does not eliminate the possibility of generating false $\mathcal{U}$ -repairs. Suppose that, in our example, we decide to eliminate $C \sqsubseteq \perp$ first because it is a root unwanted axiom. It is easily verified that the $(C \sqsubseteq \perp)$ -repairs for Γ are $\Gamma_3 = \{2, 3, 4\}$ and $\Gamma_4 = \{1, 3, 4\}$. Having eliminated $C \sqsubseteq \perp$, we then proceed to eliminate the remaining unwanted axiom $F \sqsubseteq \perp$ from both Γ_3 and Γ_4 . It is easily verified that $\Gamma_3 \not\models F \sqsubseteq \perp$, but that

$\Gamma_4 \models F \sqsubseteq \perp$. So we leave Γ_3 unchanged, but we need to generate the $(F \sqsubseteq \perp)$ -repairs of Γ_4 . They are $\{3, 4\}$ and $\{1, 4\}$. The candidate $\mathcal{U}$ -repairs for Γ are therefore Γ_3 , $\{3, 4\}$, and $\{1, 4\}$. And as can be verified, Γ_3 and $\{1, 4\}$ are both $\mathcal{U}$ -repairs for Γ , but $\{3, 4\}$ is not. As we have noted, it is possible to recognise $\{3, 4\}$ as a false $\mathcal{U}$ -repair since it is a subset of one of the $\mathcal{U}$ -repairs.

3 ROOT JUSTIFICATIONS

We now briefly discuss some preliminary work on an alternative approach to ontology repair. The key difference is to deal with unwanted axioms *simultaneously*, rather than sequentially. The basic notion we need is that of a root justification. Given a Tbox Γ and a set of unwanted axioms $\mathcal{U}$, a set RJ is a $\mathcal{U}$ -root justification for Γ if and only if it is a $\mathcal{U}$ -justification for Γ for some $U \in \mathcal{U}$ (i.e. $RJ \in \mathcal{J}(\mathcal{U})$), and there is no $J \in \mathcal{J}(\mathcal{U})$ such that $J \subset RJ$. We denote the set of $\mathcal{U}$ -root justifications for Γ by $\mathcal{RJ}_\Gamma(\mathcal{U})$. As we have seen, for our example the set of all $\mathcal{U}$ -justifications is $\mathcal{J}_\Gamma(\mathcal{U}) = \{\{1, 3\}, \{1, 2, 4\}, \{1, 2\}\}$, and therefore the set of $\mathcal{U}$ -root justifications for Γ is $\mathcal{RJ}_\Gamma(\mathcal{U}) = \{\{1, 2\}, \{1, 3\}\}$.

The significance of root justifications is that they can be used to generate precisely the $\mathcal{U}$ -repairs for Γ , in the same way in which $\mathcal{U}$ -repairs are generated from justifications for a single unwanted axiom. A subset D of Γ is a $\mathcal{U}$ -diagnosis for Γ if and only if $D \cap RJ \neq \emptyset$ for every $RJ \in \mathcal{RJ}_\Gamma(\mathcal{U})$. D is a *minimal* $\mathcal{U}$ -diagnosis for Γ if and only if there is no $\mathcal{U}$ -diagnosis D' (for Γ) such that $D' \subset D$. The set of minimal $\mathcal{U}$ -diagnoses for Γ is denoted by $\mathcal{D}_\Gamma(\mathcal{U})$. We then have the following theorem showing that the $\mathcal{U}$ -repairs for Γ can be obtained from the $\mathcal{U}$ -diagnoses for Γ :

Theorem 2 $\mathcal{R}_\Gamma(\mathcal{U}) = \{\Gamma \setminus D \mid D \in \mathcal{D}_\Gamma(\mathcal{U})\}$.

For our example we have already seen that $\mathcal{RJ}_\Gamma(\mathcal{U}) = \{\{1, 2\}, \{1, 3\}\}$. From this it follows that $\mathcal{D}_\Gamma(\mathcal{U}) = \{\{1\}, \{2, 3\}\}$ and therefore, as indicated by the theorem, that $\mathcal{R}_\Gamma(\mathcal{U}) = \{\{2, 3, 4\}, \{1, 4\}\}$.

We have implemented a Protégé 4 plugin³ for computing root justifications for sets of unwanted axioms (<http://ksg.meraka.org.za/~kmoodley/protege>). We are extending the plugin to compute the $\mathcal{U}$ -repairs. The next step will be to compare this approach to ontology repair with both the naïve sequential approach described above, as well as the improved sequential method which uses root unwanted axioms to determine the sequence in which unwanted axioms are eliminated.

REFERENCES

- [1] F. Baader and B. Hollunder, 'Embedding Defaults into Terminological Knowledge Representational Formalisms', *J. of Automated Reasoning*, **14**, 149–180, (1995).
- [2] F. Baader and W. Nutt, 'Basic description logics', in *The Description Logic Handbook: (2nd Ed.)*, eds., F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, Cambridge University Press, (2007).
- [3] V. Haarslev and R. Möller, 'Racer system description', in *IJCAR 2001*, eds., R. Goré, A. Leitsch, and T. Nipkow, volume LNAI 2100, (2001).
- [4] M. Horridge, J. Bauer, B. Parsia, and U. Sattler, 'Understanding Entailments in OWL', in *Proc. of OWLED*, (2008).
- [5] R. Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence*, **32**, 57–95, (1987).
- [6] S. Schlobach, 'Diagnosing terminologies', in *Proc. of AAAI05*, pp. 670–675, (2005).
- [7] D. Tsarkov and I. Horrocks, 'Efficient reasoning with range and domain constraints', in *Proceedings of DL*, pp. 41–50, (2004).

³ <http://protege.stanford.edu>

A Contextual Approach to Detection of Conflicting Ontologies

Michael Chan¹ and Jos Lehmann² and Alan Bundy³

1 INTRODUCTION

The knowledge represented in an ontology can be regarded as merely a perception from a particular perspective – whether it be that of the modeler or of an autonomous agent. Such interpretation is inline with the representation of ontological knowledge in a context, which is often regarded as a subtheory about the world for a particular situation. Our primary interest is to automatically evolve ontologies by diagnosing and repairing ontological faults, so we will describe a preliminary investigation into the detection of conflicts between ontologies by means of analysing relations between contexts. The presentation of the analysis is based on an example of a physics paradox, caused by a contradiction between the predictive theory and sensory data.

Suppose a bouncy ball B is suspended at a height above ground and a student, who believes that the total energy (TE) of an object is always defined as the summation of only the kinetic energy (KE) and potential energy (PE), is to predict TE of the ball when it impacts with the ground. Sensory data shows that both the velocity and the height can be deduced to be zero at that moment. This causes a contradiction because the TE at the end of the drop, $TE(B, End(Drop))$, calculated using the sensory data is zero, whereas the predicted value is positive by the law of energy conservation. The paradox arises from the (wrong) idealisation of the ball as a particle without extent, so the contribution of elastic energy to TE is neglected. This is the *bouncing-ball paradox*.

A natural representation of the paradox is to encode the predictive theory in one ontology (O_t) and the sensory data in another (O_s), letting each be a separate ontology and be locally consistent. The theoretical ontology contains the relevant physics laws, including the theoretical definitions of TE , KE , PE , etc. To give an accurate representation of O_s , in contrast, is to not assert the values of final velocity and height, but to assert the values from the raw data. In essence, the student's calculations of final velocity and height are based on the raw data collected from the experiment and are, therefore, deduced. However, physics laws and relevant definitions are defined in terms of basic properties such as velocities and heights and do not directly refer to specific attributes of an experiment. Thus, it is essential to bridge this gap, created by the heterogeneity of signatures, so that the relevant terms are properly related across the ontologies.

2 HETEROGENEOUS SIGNATURES

Requiring O_t and O_s to share a set of signature elements and contain relevant definitions can avoid numerous problems associated with the reasoning and the representation, but at the cost of decreased accuracy and generality. For more accurate and flexible representations, we handle the case in which the ontologies do not share a common signature. Let O_t contain the law of energy conservation, relevant definitions, and claims about the initial state of the ball:

$$\begin{aligned} Ax(\mathcal{T}(O_t)) &::= \{ \forall p:Part, t_i, t_j: Mom. TE(p, t_i) = TE(p, t_j), \\ &\forall p:Part, t: Mom. TE(p, t) = KE(p, t) + PE(p, t), \\ &\forall p:Part, t: Mom. KE(p, t) ::= \frac{Mass(p, t).Vel(p, t)^2}{2}, \\ &\forall p:Part, t: Mom. PE(p, t) ::= Mass(p, t).G.Height(p, t) \} \end{aligned}$$

$$\begin{aligned} Ax(\mathcal{A}(O_t)) &::= \{ Vel(B, Start(Drop)) = 0, \\ &Height(B, Start(Drop)) > 0, \dots \} \cup Ax(\mathcal{T}(O_t)) \end{aligned}$$

where $Part$ and Mom respectively denote the sets of particles and moments; $Ax(\mathcal{T}(O))$ and $Ax(\mathcal{A}(O))$ denote the axioms of the TBox and ABox of the ontology O , respectively. We adopt description logic's distinction between TBox and ABox even though we work with higher-order logic. As will be explained later, this distinction provides several technical benefits. As with O_s , we shall augment the background story of the paradox by assuming that the experiment involves only shooting a series of high-speed photos of the ball while it drops:

$$\begin{aligned} Ax(\mathcal{A}(O_s)) &::= \{ Posn(B, Photo(B, End(Drop) - \Delta)) = 0, \\ &Posn(B, Photo(B, End(Drop))) = 0, \dots \} \end{aligned}$$

where $Posn(B, Photo) = p$ means that the position of the ball B in the photograph $Photo$ is at position p . In the current setting, we can assume $Posn$ to return the 1-D position according to a fixed reference, e.g., a vertical ruler; $Photo(O, T)$ returns the photo of object O taken at time T .

The above configuration requires a significantly different approach for conflict diagnosis because the knowledge represented in O_t and O_s is insufficient to relate the terms in O_t to $Posn$ in O_s . Such asymmetry renders the derivation of a contradiction impossible, but may be tackled, e.g., by using McCarthy's notion of *lifting axioms* described in his work on the formalisation of contexts [3] and in Guha's work on microtheories [2].

2.1 Lifting Axioms

Lifting axioms are rules that help bridge across individual contexts, enabling terms from one context to be translated

¹ University of Edinburgh, UK; M.Chan@ed.ac.uk

² University of Edinburgh, UK; J.Lehmann@inf.ed.ac.uk

³ University of Edinburgh, UK; A.Bundy@ed.ac.uk

into another. Using McCarthy's syntax, we may need another ontology, O_b , containing information about relationships between the terms in $\mathcal{T}(O_t)$ and those in O_s in order to bridge across them. Even O_b connects with O_t and O_s , an inconsistency is avoided because the value assertions in the ABoxes are not included in O_b , which eliminates the potential problem of merging conflicting value assertions. For the described O_t and O_s , the axioms of $\mathcal{T}(O_b)$ can be:

$$Ax(\mathcal{T}(O_b)) ::= \{ \forall p:Part, t:Mom. O_t.Vel(p, t) = \quad (1)$$

$$(O_s.Posn(p, Photo(p, t - \Delta)) - O_s.Posn(p, Photo(p, t))) / (t - (t - \Delta)),$$

$$\forall p:Part, t:Mom. O_t.Height(p, t) = O_s.Posn(p, t), \quad (2)$$

$$\forall p:Part, t_i, t_j:Mom. O_t.TE(p, t_i) = O_t.TE(p, t_j), \dots \} \quad (3)$$

The term $c.p$ is our shorthand for McCarthy's term *value*(c, p), which designates the value of the term p in a context (in our case, an ontology) c . Thus, (1) expresses that the value of $Vel(p, t)$ in O_t is equal to the difference between the positions returned by $Posn$ in O_s given two photos, each taken at the beginning and the end of the corresponding interval, divided by the length of the interval. Similarly, (2) means that the value of $Height(p, t)$ in O_t is the same as the value of $Posn(p, t)$ in O_s .

As one would expect, lifting axioms can also be used to relate terms in the simple setup of ontologies that share the same signature. For example, if O_s shares the same signature as O_t and has a theory over the domain containing only B and $End(Drop)$:

$$Ax(\mathcal{T}(O_s)) ::= \{ TE(B, End(Drop)) = KE(B, End(Drop)) + PE(B, End(Drop)), \dots \}$$

The corresponding O_b will not contain (1) or (2), but contain the following additional lifting axioms:

$$Ax(\mathcal{T}(O_b)) ::= \{ O_t.TE(B, End(Drop)) = O_s.TE(B, End(Drop)), \dots \}$$

3 CONFLICT DETECTION

We adopt a similar approach to detecting a conflict between ontologies represented using contexts as that described in [1], i.e. logically derive formulae from the ontologies that imply a derivable contradiction. Since the representation of ontologies as contexts is more complex than that using object-level logic, care is required to reason with both the meta- and object-levels. Before deriving the trigger formulae from the ontologies, each axiom in O_t and O_s need to be syntactically modified by renaming every term in O_t and O_s , so that the end results are guaranteed to not share any part of the signature. Consequently, O_b is modified accordingly as it specifies the relationships between the terms in the working ontologies. The purpose of the renaming is to avoid an inconsistency from arising when parts of the ontologies are merged, which will be later described. For some ontology O_i , the axioms of the renamed ontology O_i' are:

$$Ax(O_i') ::= \{ \phi\{tm^i/tm\} \mid \phi \in Ax(O_i), tm \in \phi \}$$

where $\phi\{tm^i/tm\}$ denotes that occurrences of term tm in the formula ϕ are renamed to tm^i ; $tm \in \phi$ means that tm is a term in the formula ϕ . The resulting O_t' , O_s' , and O_b' for O_t , O_s , and O_b , respectively, are therefore:

$$Ax(\mathcal{T}(O_t')) ::= \{ \forall p:Part, t_i, t_j:Mom. TE^t(p, t_i) = TE^t(p, t_j), \forall p:Part, t:Mom. TE^t(p, t) = KE^t(p, t) + PE^t(p, t), \dots \}$$

$$Ax(\mathcal{A}(O_s')) ::= \{ Posn^s(B, Photo^s(B, End^s(Drop) - \Delta)) = 0, Posn^s(B, Photo^s(B, End^s(Drop))) = 0 \}$$

$$Ax(O_b') ::= \{ \forall p:Part, t:Mom. O_t.Vel^t(p, t) = (O_s.Posn^s(p, Photo^s(p, t - \Delta)) - O_s.Posn^s(p, Photo^s(p, t))) / (t - (t - \Delta)), \dots \}$$

For the sake of symmetry, terms in both O_t and O_s are renamed.

Based on the trigger formulae designed for the *Where's my stuff* (WMS) ontology repair plan [1], a conflict between O_t and O_s through lifting axioms is detected if at least two of the following three are matched:

$$O_t \vdash stuff(\vec{s}) = v_1 \quad (4)$$

$$O_s \vdash stuff(\vec{s}) = v_2 \quad (5)$$

$$O_b \vdash o.stuff(\vec{s}) = \psi, Th(\{decontext(o.stuff(\vec{s}) = \psi)\} \cup \quad (6)$$

$$Ax(\mathcal{A}(O_t')) \cup Ax(\mathcal{A}(O_s')) \vdash stuff(\vec{s}) = v_b$$

where $O_b \vdash o.stuff(\vec{s}) = \psi$ means that the term $stuff(\vec{s})$ in ontology o can be expressed as ψ in O_b ; $decontext(\phi)$ decontextualises the formula ϕ such that every term in ϕ is considered to reside in the same context, i.e. $decontext(o_1.f = o_2.g)$ gives $f = g$. With WMS, conflict is detected if only (4) and (5) are matched and that $O_t \vdash v_1 \neq v_2$. The coverage of this trigger is somewhat limited because, for example, $stuff(B, End(Drop)) = v$ cannot be deduced in O_s alone if $stuff$ is not in the signature of O_s . The WMS trigger formulae can be augmented with (6), such that any two of (4), (5), and (6), and that $O_t \vdash v_1 \neq v_2 (\neq v_b)$, depending on the matching formulae, can trigger repair. Note that the resulting merge of the two ABoxes in (6) is guaranteed to be consistent as O_t and O_s no longer potentially share signature elements, due to the renaming by $\{tm^t/tm \mid tm \in \phi\}$ and $\{tm^s/tm \mid tm \in \phi\}$ to every term in each respective ontology.

A conflict can be detected in the bouncing-ball paradox represented using the lifting axioms in (1) and onwards: (4) and (6) can be matched by the substitution $\{stuff/TE^t, \vec{s}/(B, End(Drop)), o/O_t, v_1/x, x > 0, v_b/0\}$ and substituting ψ for the sum of KE and PE , expressed in respect to the terms in O_s :

$$0.5 \cdot Mass^s(B, End^s(Drop)).((Posn^s(B, Photo^s(B, End^s(Drop) - \Delta)) - Posn^s(B, Photo^s(B, End^s(Drop)))) / (End^s(Drop) - (End^s(Drop) - \Delta)))^2 + \dots$$

Clearly, the complexity of the detection mechanism presented is significantly higher than that presented in [1]. One obvious challenge is to reason with knowledge in both meta- and object-levels, i.e. that in O_b and O_t and/or O_s . That said, detecting ontological conflicts as contextual ones enables higher generality, thus accuracy, in the modelling and representation.

ACKNOWLEDGEMENTS

The research reported in this paper was supported by EPSRC grant EP/E005713/1.

REFERENCES

- [1] A. Bundy and M. Chan. Towards ontology evolution in physics. In W. Hodges, editor, *Procs. Wollic 2008*. LNCS, Springer-Verlag, July 2008.
- [2] Ramanathan Guha. *Contexts: a formalization and some applications*. PhD thesis, Stanford, CA, USA, 1992.
- [3] J. McCarthy and S. Buvac. Formalizing context (expanded notes). *Computing natural language*, 81:13–50, 1998.

