ADVANCES IN ONTOLOGIES

Proceedings of the Eighth Australasian Ontology Workshop, Sydney, Australia

4 December 2012

Editors

Aurona Gerber
Kerry Taylor
Tommie Meyer
Mehmet Orgun

Preface

The Australasian Ontology Workshop series was initiated in 2005, and AOW 2012 is the eighth in the series. In 2012, AOW was held on the 4th of December 2012 in Sydney, Australia. Like most of the previous events AOW 2012 was held as a workshop of the Australasian Joint Conference on Artificial Intelligence celebrating its 25th anniversary as AI2012.

Out of papers submitted, we accepted 5 full papers and 4 short papers on the basis of three or four reviews submitted by our Program Committee of international standing. The submissions covered an interesting balance of topics with papers on fundamental research in ontologies, to ontology applications. We were pleased to note that we again attracted international authors.

As in previous years, an award of $250 AUD was made available for the best paper, sponsored this year by CAIR [1] (the Centre for Artificial Intelligence Research in South Africa). In 2012 the best paper prize was awarded to Giovanni Casini and Alessandro Mosca for their paper "Defeasible reasoning in ORM2".

AOW 2012 was the last AOW in its current form. From 2013 AOW will be replaced by the Australasian Semantic Web Conference (ASWC). The 12th International Semantic Web Conference (ISWC) and the 1st Australasian Semantic Web Conference (ASWC) will be held 21-25 October 2013 in Sydney, Australia and from 2014 onwards, ASWC will be a free-standing conference.

Many individuals contributed to this workshop. We thank our contributing authors and dedicated international Program Committee for their careful reviews in a tight time frame. We also thank CAIR for sponsoring the memory keys containing the proceedings. We acknowledge the EasyChair conference management system, which was used in all stages of the paper submission and review process and also in the collection of the final camera-ready papers, as well as the Yola web authoring system for our website available at http://aow2012.yolasite.com. We hope that you found this eighth Australasian Ontology Workshop to be informative, thought provoking, and most of all, enjoyable!

Kerry Taylor (CSIRO ICT Centre, Australia) (co-chair)
Aurona J. Gerber (CAIR, South Africa) (co-chair)
Tommie Meyer (CAIR, South Africa) (co-chair)
Mehmet A. Orgun (Macquarie University, Australia) (co-chair)

Organisers of AOW 2012
December 2012

[1] http://www.cair.za.net/

Conference Organisation

Programme Chairs

Kerry Taylor (CSIRO ICT Centre, Australia)
Aurona J. Gerber (CAIR – Centre for Artificial Intelligence Research, South Africa)
Tommie Meyer (CAIR – Centre for Artificial Intelligence Research, South Africa)
Mehmet A. Orgun (Macquarie University, Australia)

Programme Committee

Franz	Baader	TU Dresden
Michael	Bain	UNSW
Arina	Britz	Meraka Institute, CSIR
Giovanni	Casini	CAIR (CSIR and UKZN)
Werner	Ceusters	SUNY at Buffalo
Michael	Compton	CSIRO
Oscar	Corcho	Universidad Politécnica de Madrid
Atilla	Elci	Süleyman Demirel University
R. Cenk	Erdur	Ege University
Peter	Fox	TWC/RPI
Manolis	Gergatsoulis	Dept. of Archive and Library Science, Ionian University
Tudor	Groza	School of ITEE, The University of Queensland
Armin	Haller	Digital Enterprise Research Institute (DERI), National University of Ireland, Galway
Knut	Hinkelmann	University of Applied Sciences Northwestern Switzerland FHNW
Bo	Hu	SAP Research
C. Maria	Keet	School of Mathematics, Statistics, and Computer Science, University of KwaZulu-Natal, South Africa
Aneesh	Krishna	Curtin University, Australia
Kevin	Lee	National ICT Australia and University of New South Wales
Laurent	Lefort	CSIRO ICT Centre
Yuan-Fang	Li	Monash University
Constantine	Mantratzis	University of Westminster
Deshendran	Moodley	University of Computer Science
Maurice	Pagnucco	The University of New South Wales
Jeff Z.	Pan	University of Aberdeen
Deborah	Richards	Macquarie University
Rolf	Schwitter	Macquarie University
Barry	Smith	SUNY Buffalo
Markus	Stumptner	University of South Australia
Vijayan	Sugumaran	School of Business Administration, Oakland University, Rochester, MI 48309, USA
Boontawee	Suntisrivaraporn	Sirindhorn International Institute of Technology
Sergio	Tessaris	Free University of Bozen - Bolzano
Nwe Ni	Tun	Research Fellow
Ivan	Varzinczak	Centre for AI Research, CSIR Meraka Institute and UKZN
Kewen	Wang	Griffith University
Levent	Yilmaz	Auburn University
Antoine	Zimmermann	École des Mines de Saint-Étienne

AOW 2012 Accepted Papers

Defeasible reasoning in ORM2

Giovanni Casini[1] and Alessandro Mosca[2]

[1] Centre for Artificial Intelligence Research, CSIR Meraka Institute and UKZN, South Africa
Email: GCasini@csir.co.za
[2] Free University of Bozen-Bolzano, Faculty of Computer Science, Italy
Email: mosca@inf.unibz.it

Abstract. The *Object Role Modeling* language (ORM2) is one of the main conceptual modeling languages. Recently, a translation has been proposed of a main fragment of ORM2 (ORM2$^{\text{zero}}$) into the description logic $\mathcal{ALCQI}$, allowing the use of logical instruments in the analysis of ORM schemas. On the other hand, in many ontological domains there is a need for the formalization of *defeasible information* and of *nonmonotonic* forms of reasoning. Here we introduce two new constraints in ORM2 language, in order to formalize defeasible information into the schemas, and we explain how to translate such defeasible information in $\mathcal{ALCQI}$.

1 Introduction

ORM2 ('Object Role Modelling 2') is a graphical fact-oriented approach for modelling, transforming, and querying business domain information, which allows for a verbalisation in a language readily understandable by non-technical users [1]. ORM2 is at the core of the OGM standard SBVR language ('Semantics of Business Vocabulary and Business Rules'), and of conceptual modelling language for database design in Microsoft Visual Studio (VS). In particular, the Neumont ORM Architect (NORMA) tool is an open source plug-in to VS providing the most complete support for the ORM2 notation.

On the other hand, in the more general field of formal ontologies in the last years a lot of attention has been dedicated to the implementations of forms of *defeasible reasoning*, and various proposals, such as [2,3,4,5,6,7,8], have been made in order to integrate nonmonotonic reasoning mechanisms into DLs.

In what follows we propose an extension of ORM2 with two new formal constraints, with the main aim of integrating a form of defeasible reasoning in the ORM2 schemas; we explain how to translate such enriched ORM2 schemas into $\mathcal{ALCQI}$ knowledge bases and how to use them to check the schema consistency and draw conclusions. In particular, the paper presents a procedure to implement a particular construction in nonmonotonic reasoning, *i.e.* Lehmann and Magidor's *Rational Closure* (RC)[9], that is known for being characterized by good logical properties and for giving back intuitive results.

2 Fact-oriented modelling in ORM2

'Fact-oriented modelling' began in the early Seventies as a conceptual modelling approach that views the world in terms of simple facts about individuals and the roles they play [1]. *Facts* are assertions that are taken to be true in the domain of interest about

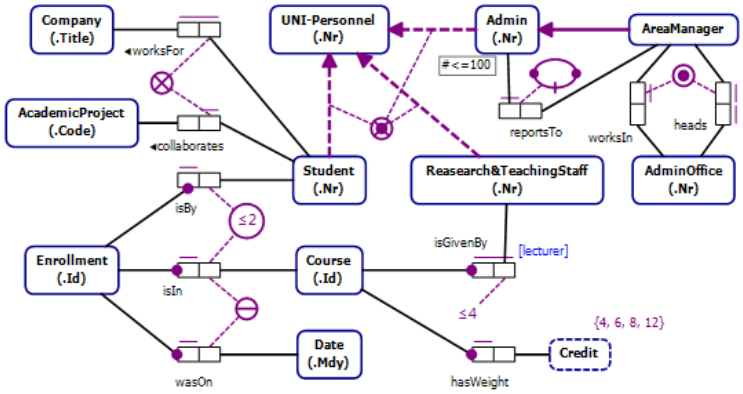

Fig. 1. A conceptual schema including an instantiation of most of the ORM2 constraints.

objects playing certain *roles* (e.g. 'Alice is enrolled in the Computer Science program'). In ORM2 one has **entities** (e.g. a person or a car) and **values** (e.g. a *character string* or a *number*). Moreover, entities and values are described in terms of the **types** they belong to, where a type (e.g. House, Car) is a set of instances. Each entity in the domain of interest is, therefore, an instance of a particular type. The roles played by the entities in a given domain are introduced by means of logical **predicates**, and each predicate has a given set of **roles** according to its arity. Each role is connected to exactly one object type, indicating that the role is played only by the (possible) instances of that type ((e.g. TYPE(isBy.Student,Student)) - notice that, unlike ER, ORM2 makes no use of 'attributes'). ORM2 also admits the possibility of making an object type out of a relationship. Once a relation has been transformed into an object type, this last is called the **objectification** of the relation.

According to the ORM2 design procedure, after the specification of the relevant **object types** (i.e. entity and value types) and predicates, the static *constraints* must be considered. The rest of this section is devoted to an informal introduction of the constraint graphical representation, together with their intended semantics. Fig. 1 shows an example of an ORM2 conceptual schema modelling the 'academic domain' (where the soft rectangles are entity types, the dashed soft rectangles are value types, and the sequences of one or more role-boxes are predicates). The example is not complete w.r.t. the set of all the ORM2 constraints but it aims at giving the feeling of the expressive power of the language. The following are among the constraints included in the schema (the syntax we devised for linearizing them is in square brackets):

1. **Subtyping** (depicted as thick solid and dashed arrows) representing 'is-a' relationships among types. A **partition**, made of a combination of an **exclusive** constraint (a circled 'X' saying that 'Research&TeachingStaff, Admin, Student are *mutually disjoint*'), and a **total** constraint (a circled dot for 'Research&TeachingStaff, Admin, Student completely *cover* their common supertype'). [O-SET$_{Tot}$({Research&TeachingStaff, Admin, Student},UNI-Personnel)]
2. An **internal frequency occurrence** saying that *if* an instance of Research&TeachingStaff plays the role of being lecturer in the relation isGivenBy, that instance can play the role at most 4 times [FREQ(isGivenBy.Research&TeachingStaff,(1,4))]. A frequency occurrence may span over more than one role, and suitable frequency *ranges* can be specified. *At*

most one cardinalities (depicted as continuos bars) are special cases of frequency occurrence called **internal uniqueness** constraints [e.g. FREQ(hasWeight.Course,(1,1))].

3. An **external frequency occurrence** applied to the roles played by Student and Course, meaning that 'Students are allowed to enrol in the same course *at most twice*'. [FREQ(isIn.Course,isBy.Student,(1,2))]
4. An **external uniqueness** constraint between the role played by Course in isIn and the role played by Date in wasOn, saying that 'For each combination of Course and Date, *at most one* Enrollment isIn that Course and wasOn that Date'. [FREQ(isIn.Course,wasOn.Date,(1,1))]
5. A **mandatory participation** constraints (graphically represented by a dot), among several other, saying that 'Each Course isGivenBy *at least one* instance of the Research&TeachingStaff type' (combinations of mandatory and uniqueness translate into *exaclty one* cardinality constraints). [MAND(isGivenBy.Research&TeachingStaff,Research&TeachingStaff)]
6. A disjunctive mandatory participation, called **inclusive-or** constraint (depicted as a circled dot), linking the two roles played by the instances of AreaManager meaning that 'Each area manager *either* works in *or* heads (or *both*)'. [MAND({worksIn.AreaManager,heads.AreaManager},AreaManager)]
7. An **object cardinality** constraint forcing the number of the Admin instances to be less or equal to 100 (**role cardinality** constraints, applied to role instances, are also part of ORM2). [O-CARD(Admin)=(0,100)]
8. An **object type value** constraint indicating which values are allowed in Credit (**role value** constraints can be also expressed to indicate which values are allowed to play a given role). [V-VAL(Credit)={4,6,8,12}]
9. An **exclusion** constraint (depicted as circled 'X') between the two roles played by the instances of Student, expressing the fact that no student can play *both* these roles. Exclusion constraint can also span over arbitrary sequences of roles. The combination of exclusion and inclusive-or constraints gives rise to **exclusive-or** constraints meaning that each instance in the attached entity type plays *exactly one* of the attached roles. Exclusion constraints, together with **subset** and **equality**, are called *set-comparison* constraints. [$\text{R-SET}_{\text{Exc}}$(worksFor.Student,collaborates.Student)]
10. A **ring** constraint expressing that the relation reportsTo is *asymmetric*. [$\text{RING}_{\text{Asym}}$(reportTo.Admin,reportTo.AreaManager)]

A comprehensive list of all the ORM2 constraints, together with their graphical representation, can be found in [1].

3 The $\mathcal{ALCQI}$ encoding of ORM2$^{\text{zero}}$

With the main aim of relying on available tools to reason in an effective way on ORM2 schemas, an encoding in the description logic $\mathcal{ALCQI}$ for which tableaux-based reasoning algorithms with a tractable computational complexity have been developed [10]. $\mathcal{ALCQI}$ corresponds to the basic DL $\mathcal{ALC}$ equipped with *qualified cardinality restrictions* and *inverse roles*, and it is a fragment of the OWL2 web ontology language (a complete introduction of the syntax and semantics of $\mathcal{ALCQI}$ can be found in [11]). We also introduce in the $\mathcal{ALCQI}$ language the expression '$C \supset D$' as an abbreviation for the expression '$\neg C \sqcup D$'.

Now, the discrepancy between ORM2 and $\mathcal{ALCQI}$ poses two main obstacles that need to be faced in order to provide the encoding. The first one, caused by the absence of n-ary relations in $\mathcal{ALCQI}$, is overcome by means of *reification*: for each relation R of

arity $n \geq 2$, a new atomic concept A_R and n functional roles $\tau(R.a_1), \ldots, \tau(R.a_n)$ are introduced. The tree-model property of $\mathcal{ALCQI}$ guarantees the *correctness* encoding w.r.t. the reasoning services over ORM2. Unfortunately, the second obstacle fixes, once for all, the limits of the encoding: $\mathcal{ALCQI}$ does not admit neither arbitrary set-comparison assertions on relations, nor external uniqueness or uniqueness involving more than one role, or arbitrary frequency occurrence constraints. In other terms, it can be proven that $\mathcal{ALCQI}$ is strictly contained in ORM2. The analysis of this inclusion thus led to identification of the fragment called $\text{ORM2}^{\text{zero}}$ which is maximal with respect to the expressiveness of $\mathcal{ALCQI}$, and still expressive enough to capture the most frequent usage patterns of the conceptual modelling community. Let $\text{ORM2}^{\text{zero}} = \{\text{TYPE}, \text{FREQ}^-, \text{MAND}, \text{R-SET}^-, \text{O-SET}_{\text{Isa}}, \text{O-SET}_{\text{Tot}}, \text{O-SET}_{\text{Ex}}, \text{OBJ}\}$ be the fragment of ORM2 where: (i) FREQ^- can only be applied to single roles, and (ii) R-SET^- applies either to entire relations of the same arity or to two single roles. The encoding of the semantics of $\text{ORM2}^{\text{zero}}$ shown in table 1 is based on the $\mathcal{S}^{\mathcal{ALCQI}}$ signature made of: (i) A set $E_1, E_2, \ldots, E_n$ of concepts for *entity types*; (ii) a set $V_1, V_2, \ldots, V_m$ of concepts for *value types*; (iii) a set $A_{R_1}, A_{R_2}, \ldots, A_{R_k}$ of concepts for objectified *n*-ary *relations*; (iv) a set $D_1, D_2, \ldots, D_l$ of concepts for *domain symbols*; (v) $1, 2, \ldots, n_{max} + 1$ roles. Additional *background axioms* are needed here in order to: (i) force the interpretation of the $\mathcal{ALCQI}$ knowledge base to be correct w.r.t. the corresponding ORM2 schema, and (ii) guarantee that that any model of the resulting $\mathcal{ALCQI}$ can be 'un-reified' into a model of original $\text{ORM2}^{\text{zero}}$ schema.
The correctness of the introduced encoding is guaranteed by the following theorem (whose complete proof is available at [12]):

Theorem 1. *Let Σ^{zero} be an $\text{ORM2}^{\text{zero}}$ conceptual schema and $\Sigma^{\mathcal{ALCQI}}$ the $\mathcal{ALCQI}$ KB constructed as described above. Then an object type O is consistent in Σ^{zero} if and only if the corresponding concept O is satisfiable w.r.t. $\Sigma^{\mathcal{ALCQI}}$.*

Let us conclude this section with some observation about the complexity of reasoning on ORM2 conceptual schemas, and taking into account that all the reasoning tasks for a conceptual schema can be reduced to object type consistency. Undecidability of the ORM2 object type consistency problem can be proven by showing that arbitrary combinations of subset constraints between n-ary relations and uniqueness constraints over single roles are allowed [13]. As for $\text{ORM2}^{\text{zero}}$, one can conclude that object type consistency is EXPTIME-complete: the upper bound is established by reducing the $\text{ORM2}^{\text{zero}}$ problem to concept satisfiability w.r.t. $\mathcal{ALCQI}$ KBs (which is known to be EXPTIME-hard) [14], the lower bound by reducing concept satisfiability w.r.t. $\mathcal{ALC}$ KBs (which is known to be EXPTIME-complete) to object consistency w.r.t. $\text{ORM2}^{\text{zero}}$ schemas [15]. Therefore, we obtain the following result:

Theorem 2. *Reasoning over $\text{ORM2}^{\text{zero}}$ schemas is* EXPTIME*-complete.*

4 Rational Closure in $\mathcal{ALCQI}$

Now we briefly present the procedure to define the analogous of RC for the DL language $\mathcal{ALCQI}$. A more extensive presentation of such a procedure can be found in [4]: it is defined for $\mathcal{ALC}$, but it can be applied to $\mathcal{ALCQI}$ without any modifications. RC is one of the main construction in the field of nonmonotonic logics, since it has a solid

Table 1. $\mathcal{ALCQI}$ encoding.

Background domain axioms:	$E_i \sqsubseteq \neg(D_1 \sqcup \cdots \sqcup D_l)$ for $i \in \{1, \ldots, n\}$ $V_i \sqsubseteq D_j$ for $i \in \{1, \ldots, m\}$, and some j with $1 \leq j \leq l$ $D_i \sqsubseteq \sqcap_{j=i+1}^{l} \neg D_j$ for $i \in \{1, \ldots, l\}$ $\top \sqsubseteq A_{\top_1} \sqcup \cdots \sqcup A_{\top_{n_{max}}}$ $\top \sqsubseteq (\leq 1 i.\top)$ for $i \in \{1, \ldots, n_{max}\}$ $\forall i.\bot \sqsubseteq \forall i+1.\bot$ for $i \in \{1, \ldots, n_{max}\}$ $A_{\top_n} \equiv \exists 1.A_{\top_1} \sqcap \cdots \sqcap \exists n.A_{\top_1} \sqcap \forall n+1.\bot$ for $n \in \{2, \ldots, n_{max}\}$ $A_R \sqsubseteq A_{\top_n}$ for each atomic relation *R* of arity *n* $A \sqsubseteq A_{\top_1}$ for each atomic concept *A*
$\mathsf{TYPE}(R.a, O)$	$\exists \tau(R.a)^{-}.A_R \sqsubseteq O$
$\mathsf{FREQ}^{-}(R.a, \langle \mathsf{min}, \mathsf{max} \rangle)$	$\exists \tau(R.a)^{-}.A_R \sqsubseteq \geq \mathsf{min}\ \tau(R.a)^{-}.A_R \sqcap \leq \mathsf{max}\ \tau(R.a)^{-}.A_R$
$\mathsf{MAND}(\{R^1.a_1, \ldots, R^1.a_n, \ldots, R^k.a_1, \ldots, R^k.a_m\}, O)$	$O \sqsubseteq \exists \tau(R^1.a_1)^{-}.A_{R^1} \sqcup \cdots \sqcup \exists \tau(R^1.a_n)^{-}.A_{R^1} \sqcup \cdots \sqcup \exists \tau(R^k.a_1)^{-}.A_{R^k} \sqcup \cdots \sqcup \exists \tau(R^k.a_m)^{-}.A_{R^k}$
$^{(A)}$ $\mathsf{R\text{-}SET}^{-}_{\mathsf{Sub}}(A, B)$	$A_R \sqsubseteq A_S$ $\quad ^{(A)}\ A = \{R.a_1, \ldots, R.a_n\},\ B = \{S.b_1, \ldots, S.b_n\}$
$^{(A)}$ $\mathsf{R\text{-}SET}^{-}_{\mathsf{Exc}}(A, B)$	$A_R \sqsubseteq A_{\top_n} \sqcap \neg A_S$
$^{(B)}$ $\mathsf{R\text{-}SET}^{-}_{\mathsf{Sub}}(A, B)$	$\exists \tau(R.a_i)^{-}.A_R \sqsubseteq \exists \tau(S.b_j)^{-}.A_S$ $\quad ^{(B)}\ A = \{R.a_i\},\ B = \{S.b_j\}$
$^{(B)}$ $\mathsf{R\text{-}SET}^{-}_{\mathsf{Exc}}(A, B)$	$\exists \tau(R.a_i)^{-}.A_R \sqsubseteq A_{\top_n} \sqcap \neg \exists \tau(S.b_j).A_S$
$\mathsf{O\text{-}SET}_{\mathsf{Isa}}(\{O_1, \ldots, O_n\}, O)$	$O_1 \sqcup \cdots \sqcup O_n \sqsubseteq O$
$\mathsf{O\text{-}SET}_{\mathsf{Tot}}(\{O_1, \ldots, O_n\}, O)$	$O \sqsubseteq O_1 \sqcup \cdots \sqcup O_n$
$\mathsf{O\text{-}SET}_{\mathsf{Ex}}(\{O_1, \ldots, O_n\}, O)$	$O_1 \sqcup \cdots \sqcup O_n \sqsubseteq O$ and $O_i \sqsubseteq \sqcap_{j=i+1}^{n} \neg O_j$ for each $i = 1, \ldots, n$
$\mathsf{OBJ}(R, O)$	$O \equiv A_R$

logical characterization, it generally maintains the same complexity level of the underlying monotonic logic, and it does not give back counter-intuitive conclusions; its main drawback is in its inferential weakness, since there could be desirable conclusions that we won't be able to draw (see example 2 below).

As seen above, each ORM2$^{\mathsf{zero}}$schema can be translated into an $\mathcal{ALCQI}$ TBox. A TBox $\mathcal{T}$ for $\mathcal{ALCQI}$ consists of a finite set of general inclusion axioms (GCIs) of form $C \sqsubseteq D$, with C and D concepts. Now we introduce also a new form of information, the *defeaible inclusion axioms* $C \sqsubsetneq D$, that are read as 'Typically, an individual falling under the concept C falls also under the concept D'. We indicate with $\mathcal{B}$ the finite set of such inclusion axioms.

Example 1. Consider a modification of the classical 'penguin example', with the concepts P, B, F, I, Fi, W respectively read as 'penguin', 'bird', 'flying', 'insect', 'fish', and 'have wings', and a role $Prey$, where a role instantiation (a, b):$Prey$ read as 'a preys for b'. We can define a defeasible KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ with $\mathcal{T} = \{P \sqsubseteq B, I \sqsubseteq \neg Fi\}$ and $\mathcal{B} = \{P \sqsubsetneq \neg F,\ B \sqsubsetneq F,\ P \sqsubsetneq \forall Prey.Fi, B \sqsubsetneq \forall Prey.I, B \sqsubsetneq W\}$.

In order to define the rational closure of a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$, we must first of all transform the knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ into a new knowledge base $\langle \Phi, \Delta \rangle$, s.t. while $\mathcal{T}$ and $\mathcal{B}$ are sets of inclusion axioms, Φ and Δ are simply sets of concepts. Then, we shall use the sets $\langle \Phi, \Delta \rangle$ to define a nonmonotonic consequence relation that models the rational closure. Here we just present the procedure, referring to [4] for a more in-depth explanation of the various steps.

Transformation of $\langle \mathcal{T}, \mathcal{B} \rangle$ into $\langle \Phi, \Delta \rangle$. Starting with $\langle \mathcal{T}, \mathcal{B} \rangle$, we apply the following steps.

Step 1. Define the set representing the *strict form* of the set $\mathcal{B}$, *i.e.* the set $\mathcal{B}^{\sqsubseteq} = \{C \sqsubseteq D \mid C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{B}\}$, and define a set $\mathfrak{A}_{\mathcal{B}}$ as the set of the antecedents of the conditionals in $\mathcal{B}$, *i.e.* $\mathfrak{A}_{\mathcal{B}} = \{C \mid C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{B}\}$.

Step 2. We determine an *exceptionality ranking* of the sequents in $\mathcal{B}$ using the set of the antecedents $\mathfrak{A}_{\mathcal{B}}$ and the set $\mathcal{B}^{\sqsubseteq}$.

Step 2.1. A concept is considered *exceptional* in a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ only if it is classically negated (*i.e.* we are forced to consider it empty), that is, C is exceptional in $\langle \mathcal{T}, \mathcal{B} \rangle$ only if

$$\mathcal{T} \cup \mathcal{B}^{\sqsubseteq} \models \top \sqsubseteq \neg C$$

where $\models$ is the classical consequence relation associated to $\mathcal{ALCQI}$. If a concept is considered exceptional in $\langle \mathcal{T}, \mathcal{B} \rangle$, also all the defeasible inclusion axioms in $\mathcal{B}$ that have such a concept as antecedent are considered exceptional. So, given a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ we can check which of the concepts in $\mathfrak{A}_{\mathcal{B}}$ are exceptional (we indicate the set containing them as $E(\mathfrak{A}_{\mathcal{B}})$), and consequently which of the axioms in $\mathcal{B}$ are exceptional (the set $E(\mathcal{B}) = \{C \mathrel{\tilde{\sqsubseteq}} D \mid C \in E(\mathfrak{A}_{\mathcal{B}})\}$).

So, given a knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ we can construct iteratively a sequence $\mathcal{E}_0, \mathcal{E}_1, \ldots$ of subsets of $\mathcal{B}$ in the following way:

- $\mathcal{E}_0 = \mathcal{B}$
- $\mathcal{E}_{i+1} = E(\mathcal{E}_i)$

Since $\mathcal{B}$ is a finite set, the construction will terminate with an empty set ($\mathcal{E}_n = \emptyset$ for some n) or a fixed point of E.

Step 2.2 Using such a sequence, we can define a ranking function r that associates to every axiom in $\mathcal{B}$ a number, representing its level of exceptionality:

$$r(C \mathrel{\tilde{\sqsubseteq}} D) = \begin{cases} i & \text{if } C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{E}_i \text{ and } C \mathrel{\tilde{\sqsubseteq}} D \notin \mathcal{E}_{i+1} \\ \infty & \text{if } C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{E}_i \text{ for every } i\,. \end{cases}$$

Here we shall assume that every concept has a finite ranking value, and we shall deal with the possible occurrence of some concept with ∞ as ranking value in the following section.

Step 3. Now we build a new formalization of the information contained in the knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$, translating each of the two sets of axioms into two sets of concepts, Φ and Δ respectively. The set Φ will simply correspond to the *materialization* of the inclusion axioms, *i.e.* the concepts translating the axioms.

$$\Phi = \{C \sqsupset D \mid C \sqsubseteq D \in \mathcal{T}\}$$

In order to define the set Δ, given the rank value of the sequents in $\mathcal{B}$, we construct a set of *default concepts* $\Delta = \{\delta_0, \ldots, \delta_n\}$ (with n the highest rank-value in $\mathcal{B}$), with

$$\delta_i = \bigsqcap \{C \sqsupset D \mid C \mathrel{\tilde{\sqsubseteq}} D \in \mathcal{B} \text{ and } r(C \mathrel{\tilde{\sqsubseteq}} D) \geq i\}\,.$$

Hence we substitute the conceptual system $\langle \mathcal{T}, \mathcal{B} \rangle$ with the pair $\langle \Phi, \Delta \rangle$, where Φ and Δ are sets of concepts, the former containing concepts to be considered valid for every individual of the domain, the latter containing concepts to be considered *defeasibly* valid, *i.e.* we apply such default concepts to an individual only if they are consistent with the information in our knowledge base. It is not difficult to see that the concepts in Δ are linearly ordered by $\models$, that is, for every δ_i, $0 \leq i < n-1$, $\models \delta_i \sqsubseteq \delta_{i+1}$.

Rational Closure. Consider now $\Phi = \{C_1 \sqsupset D_1, \ldots, C_m \sqsupset D_m\}$ and $\Delta = \{\delta_0, \ldots, \delta_n\}$. We define a nonmonotonic consequence relation between the concepts $\mathrel{|\!\sim_{\langle \Phi, \Delta \rangle}}$ that determines what presumably follows from a finite set of concepts Γ. Simply, a concept D is a defeasible consequence of Γ if it classically follows from Γ, the strict information contained in the knowledge base (*i.e.* Φ), and the first default concept δ_i that in the sequence $\langle \delta_0, \ldots, \delta_n \rangle$ results classically consistent with the rest of the premises.

Definition 1. *$\Gamma|\!\sim_{\langle\Phi,\Delta\rangle} D$ iff $\models \bigsqcap\Gamma \sqcap \bigsqcap\Phi \sqcap \delta_i \sqsubseteq D$, where δ_i is the first $(\Gamma \cup \Phi)$-consistent formula [3] of the sequence $\langle\delta_0, \ldots, \delta_n\rangle$.*

You can find in [4] an explanation of why the above procedure for DL corresponds to the rational closure defined by Lehmann and Magidor for propositional languages, and satisfies the DL translation of the basic properties characterizing rational consequence relations.

Proposition 1 ([4], Proposition 4). *$|\!\sim_{\langle\Phi,\Delta\rangle}$ is a consequence relation containing $\mathcal{K} = \langle\mathcal{T}, \mathcal{B}\rangle$ and satisfying the properties of the rational consequence relations.*

Moreover, as deciding entailment in $\mathcal{ALCQI}$ is EXPTIME-complete (see Theorem 2), and since the decidability problem for the rational closure is reducible to a finite number of decision w.r.t. the classical $\mathcal{ALCQI}$ consequence relation, we obtain immediately that

Proposition 2. *Deciding $C|\!\sim_{\langle\tilde{\mathcal{T}},\tilde{\Delta}\rangle} D$ in $\mathcal{ALCQI}$ is an EXPTIME-complete problem.*

Example 2. Consider the KB of Example 1. Hence, we start with $\mathcal{K} = \langle\mathcal{T}, \mathcal{B}\rangle$. The strict form of $\mathcal{B}$ is $\mathcal{B}^{\sqsubseteq} = \{P \sqsubseteq \neg F, B \sqsubseteq F, P \sqsubseteq \forall Prey.Fi, B \sqsubseteq \forall Prey.I, B \sqsubseteq W\}$, with $\mathfrak{A}_{\mathcal{B}} = \{P, B\}$. Following the procedure at **Step 2**, we obtain the exceptionality ranking of the sequents: $\mathcal{E}_0 = \{P \sqsubseteq \neg F, B \sqsubseteq F, P \sqsubseteq \forall Prey.Fi, B \sqsubseteq \forall Prey.I, B \sqsubseteq W\}$; $\mathcal{E}_1 = \{P \sqsubseteq \neg F, P \sqsubseteq \forall Prey.Fi\}$; $\mathcal{E}_2 = \emptyset$. Automatically, we have the ranking values of every sequent in $\mathcal{B}$: namely, $r(B \sqsubseteq F) = r(B \sqsubseteq \forall Prey.I) = r(B \sqsubseteq W) = 0$; $r(P \sqsubseteq \neg F) = r(P \sqsubseteq \forall Prey.Fi) = 1$. From such a ranking, we obtain a set of default concepts $\Delta = \{\delta_0, \delta_1\}$, with

$$\delta_0 = (B \sqsupset F) \sqcap (B \sqsupset \forall Prey.I) \sqcap (P \sqsupset \neg F) \sqcap (P \sqsupset \forall Prey.Fi) \sqcap (B \sqsupset W)$$
$$\delta_1 = (P \sqsupset \neg F) \sqcap (P \sqsupset \forall Prey.Fi) \; .$$

Now, referring to definition 1, we can derive a series of desirable conclusions, as $\neg F|\!\sim\neg B$, , $B \wedge green|\!\sim F$, $P \wedge black|\!\sim\neg F$, $P|\!\sim\forall Prey.\neg I$. Instead, other counterintuitive connections are not valid, such as $B \wedge \neg F|\!\sim P$, $B \wedge \neg F|\!\sim\neg P$, or $P|\!\sim F$. Here we can notice the main weakness of the Rational Closure: even if it would be intuitive to conclude that penguins have wings ($P|\!\sim W$), we cannot conclude that a class that results atypical (as penguins) cannot inherit *any* of the typical properties of its superclasses (as having wings), even if such properties are not logically connected to the ones that determine the exceptionality (not flying and eating fish).

5 Defeasible constraints for ORM2

As seen above, in order to introduce defeasible reasoning in DL we introduce the *defeasible inclusion axiom* $C \sqsubseteq D$, indicating that the elements of the concept C *typically*, but not necessarily, are elements of the concept D. We want to introduce in the ORM2$^{\text{zero}}$schemas constraints playing an analogous role, *i.e.* representing defeasible constraints in the ontological organization of a particular domain. With this goal in mind, two constraints aimed at representing forms of defeasible constraints between classes, and classes and their properties, are introduced.

– A *defeasible subclass relation*: we introduce an arrow '$\rightsquigarrow$', where '$C \rightsquigarrow D$' indicates that each element of the class C is also an element of the class D, if not informed of the contrary.

[3] That is, $\not\models \bigsqcap\Phi \sqcap \bigsqcap\Gamma \sqsubseteq \neg\delta_i$.

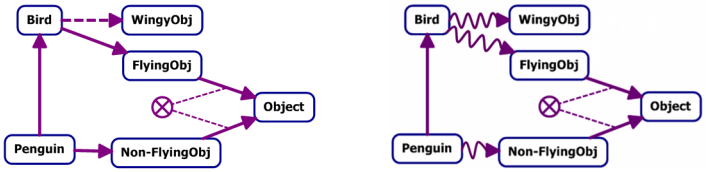

Fig. 2. Example 3, strict (left) and defeasible (right) version.

Table 2. $\mathcal{ALCQI}$ encoding of the ORM2$^{\text{zero}}$ 'Non-Flying Birds' example.

Signature:	Bird, Penguin, WingyObject, FlyingObject, $^{\text{Non}}$FlyingObject, Object
Subtyping Constraints:	FlyingObject $\sqcup$ $^{\text{Non}}$FlyingObject $\sqsubseteq$ Object
	FlyingObject $\sqsubseteq$ $\neg^{\text{Non}}$FlyingObject
	Penguin $\sqsubseteq$ Bird, Penguin $\sqsubseteq$ $^{\text{Non}}$FlyingObject
	Bird $\sqsubseteq$ WingyObject, Bird $\sqsubseteq$ FlyingObject

Example 3 (Defeasible subclass relation). Consider figure 2. The schema on the left represents in ORM2 the classic penguin example: penguins are birds and do not fly (the class Penguin is a subclass, respectively, of the classes Birds and Non-FlyingObj), while birds fly and have wings (the class Birds is a subclass, respectively, of the classes FlyingObj and WingyObj). The translation procedure into $\mathcal{ALCQI}$ gives back the TBox $\mathcal{T}$ in table 2. From $\mathcal{T}$ we can derive that the schema is inconsistent, since we have $\mathcal{T} \models \neg$Penguin, *i.e.* the concept Penguin must be empty. We can modify the knowledge base introducing defeasible information, in particular stating that birds *typically* fly and *typically* have wings, and penguins *typically* do not fly. In this way we obtain the schema on the right, and in $\mathcal{ALCQI}$ we obtain a set $\mathcal{B} = \{\text{Bird} \mathrel{|\!\sim} \text{WingyObject}, \text{Bird} \mathrel{|\!\sim} \text{FlyingObject}, \text{Penguin} \mathrel{|\!\sim} \text{NonFlyingObject}\}$, substituting the corresponding classical inclusion axioms in the TBox.

– A *defeasible mandatory participation*: we introduce a new mandatory participation constraint '$\square$', to use instead of the classically mandatory constraint '$\bullet$'. If the connection between a class C and a relation R is constrained by a constraint $\square$, we read it as 'each element of the class C participates to the relation R, if we are not informed of the contrary'.

Example 4 (Defeasible mandatory participation). Consider figure 3. The schema represents the organization of a firm: the class Manager is a subclass of the class Employee, and every employee *must* work for a project. while every project must have at least an employee working on it. The class Manager is partitioned into AreaManager and TopManager. Each top manager mandatorily manages a project. The translation procedure into $\mathcal{ALCQI}$ of the left version of the schema gives back the TBox $\mathcal{T}$ in table 3. Since managing and working for a project are not compatible roles, $\mathcal{T}$ implies that the class TopManager is empty, since a top manager would manage and would work for a project at the same time. Instead, if we declare that *typically* an employee works for a project, we can consider the top managers as exceptional kind of employees; hence we substitute the mandatory constraint between Employee and WorkFor with a defeasible constraint (*i.e.* the schema on the right in figure 3); from such

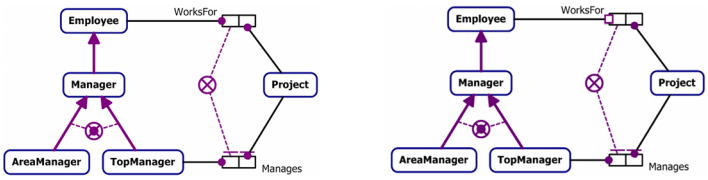

Fig. 3. Example 4, strict (left) and defeasible (right) version.

Table 3. $\mathcal{ALCQI}$ encoding of the ORM2$^{\text{zero}}$ 'Non-Managing Employees' example. Notice that: (i) according to the introduced encoding, the relations WorksFor, Manages have been reified into atomic concepts, and (ii) for the sake of clarity, we write $\exists R^{-}.C \sqsubseteq D$ instead of $C \sqsubseteq \exists R^{-}.D$).

Signature:	Employee, Manager, AreaManager, TopManager, Project, WorksFor, Manages, $A_{\top 1}$, $A_{\top 2}$ f1, f2, f3
Background axioms:	$\top \sqsubseteq A_{\top 1} \sqcup A_{\top 2}$ $\top \sqsubseteq (\leq 1 f1.\top)$, $\top \sqsubseteq (\leq 1 f2.\top)$ $\forall f1.\bot \sqsubseteq \forall f2.\bot$, $\forall f2.\bot \sqsubseteq \forall f3.\bot$ $A_{\top 2} \equiv \exists f1.A_{\top 1} \sqcap \exists f2.A_{\top 1} \sqcap \forall f3.\bot$ $\text{WorksFor} \sqsubseteq A_{\top 2}$, $\text{Manages} \sqsubseteq A_{\top 2}$ $\text{Employee} \sqsubseteq A_{\top 1}$, $\text{Manager} \sqsubseteq A_{\top 1}$, $\text{AreaManager} \sqsubseteq A_{\top 1}$, $\text{TopManager} \sqsubseteq A_{\top 1}$
Typing Constraints:	$\text{WorksFor} \sqsubseteq \exists f1^{-}.\text{Employee}$, $\text{WorksFor} \sqsubseteq \exists f2^{-}.\text{Project}$ $\text{Manages} \sqsubseteq \exists f1^{-}.\text{TopManager}$, $\text{Manages} \sqsubseteq \exists f2^{-}.\text{Project}$
Frequency Constraints:	$\exists f1^{-}.\text{Manages} \sqsubseteq = 1\, f1^{-}.\text{Manages}$
Mandatory Constraints:	$\text{Employee} \sqsubseteq \exists f1^{-}.\text{WorksFor}$ $\text{TopManager} \sqsubseteq \exists f1^{-}.\text{Manages}$ $\text{Project} \sqsubseteq \exists f2^{-}.\text{WorksFor}$ $\text{Project} \sqsubseteq \exists f2^{-}.\text{Manages}$
Exclusion Constraints:	$\exists f1^{-}.\text{WorksFor} \sqsubseteq A_{\top 2} \sqcap \neg \exists f1.\text{Manages}$
Subtyping Constraints:	$\text{Manager} \sqsubseteq \text{Employee} \sqcap (\text{AreaManager} \sqcup \text{TopeManager})$ $\text{AreaManager} \sqsubseteq \neg \text{TopeManager}$

a change we obtain a knowledge base as the one above, but with the defeasible inclusion axiom $\text{Employee} \mathrel{\sim\!\!\sqsubseteq} \exists f1^{-}.\text{WorksFor}$ instead of the axiom $\text{Employee} \sqsubseteq \exists f1^{-}.\text{WorksFor}$.

Introducing such constraints, we introduce the forms of defeasible subsumptions appropriate for modeling nonmonotonic reasoning. In particular:

- A subclass relation, as the ones in example 3, is translated into an inclusion axiom $C \sqsubseteq D$, and correspondingly we translate the defeasible connection $C \rightsquigarrow D$ into a defeasible inclusion axiom $C \mathrel{\sim\!\!\sqsubseteq} D$.
- Analogously, consider the strict form of the example 4. The mandatory participation of the class B to the role A_N is translated into the axiom $\text{B} \sqsubseteq \exists f1^{-}.\text{A}_\text{N}$. If we use the defeasible mandatory participation constraint, we simply translate the structure using the defeasible inclusion $\mathrel{\sim\!\!\sqsubseteq}$, obtaining the axiom $\text{B} \mathrel{\sim\!\!\sqsubseteq} \exists f1^{-}.\text{A}_\text{N}$.

Hence, from a ORM graph with defeasible constraints we obtain an $\mathcal{ALCQI}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$, where $\mathcal{T}$ is a standard $\mathcal{ALCQI}$ Tbox containing concept inclusion axioms $C \sqsubseteq D$, while the set $\mathcal{B}$ contains defeasible axioms of the form $C \mathrel{|\!\sim} D$. Once we have our knowledge base $\mathcal{K}$, we apply to it the procedure presented in the previous section, in order to obtain the rational closure of the knowledge base.

Consistency. In ORM2, and in conceptual modeling languages in general, the notion of consistency is slightly different from the classical form of logical consistency. That is, generally from a logical point of view a knowledge base $\mathcal{K}$ is considered inconsistent only if we can classically derive a contradiction from it; in DLs that corresponds to saying that $\mathcal{K} \models \top \sqsubseteq \bot$, *i.e.* every concept in the knowledge base results empty. Instead, dealing with conceptual modeling schemas we generally desire that our model satisfies a stronger form of consistency constraint, that is, we want that none of the classes present in the schema are forced to be empty.

Definition 2 (Strong consistency). *A TBox $\mathcal{T}$ is* strongly consistent *if none of the atomic concepts present in its axioms are forced to be empty, that is, if $\mathcal{T} \not\models \top \sqsubseteq \neg A$ for every atomic concept A appearing in the inclusion axioms in $\mathcal{T}$.*

As seen above, the introduction of defeasible constraints into ORM2$^{\text{zero}}$ allows to build schemas that in the standard notation would be considered inconsistent, but that, once introducing the defeasible constraints, allow for an instantiation such that all the classes result non-empty. Hence it is necessary to redefine the notion of consistency check in order to deal with such situations.

Such a consistency check is not problematic, since we can rely on the ranking procedure presented above. Consider a TBox $\mathcal{T}$ obtained by an ORM2$^{\text{zero}}$ schema, and indicate with $\mathcal{C}$ the set of all the atomic concepts used in $\mathcal{T}$. It is sufficient to check the exceptionality ranking of all the concepts in $\mathcal{C}$ with respect to $\mathcal{T}$: if a concept C has an exceptionality ranking $r(C) = n$, with $0 < n < \infty$, then it represents an atypical situation, an exception, but that is compatible with the information conveyed by the defeasible inclusion axioms. For example, in the above examples the penguins and the top managers would be empty classes in the classical formalization, but using the defeasible approach they result exceptional classes in our schemas, and we can consider them as non-empty classes while still considering the schema as consistent. The only case in which a class has to be considered necessarily empty, is when it has ∞ as ranking value: that means that, despite eliminating all the defeasible connections we can, such a concept still results empty. Then, the notion of strong consistency for ORM2$^{\text{zero}}$ with defeasible constraints is the following:

Definition 3 (Strong consistency with defeasible constraints). *A knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ is* strongly consistent *if none of the atomic concepts present in its axioms are forced to be empty, that is, if $r(A) \neq \infty$ for every atomic concept A appearing in the inclusion axioms in $\mathcal{K}$.*

Given a defeasible ORM2$^{\text{zero}}$schema Σ, eliminate from it all the defeasible constraints (call Σ_{strict} the resulting schema). From the procedures defined above, it is immediate to see that if Σ_{strict} is a strongly inconsistent ORM2$^{\text{zero}}$schema, in the 'classical' sense, then Σ is a strongly inconsistent defeasible schema: simply, if the negation of a concept is forced by the strict part of a schema, it will be necessarily forced at each ranking level, resulting in a ranking value of ∞.

On the other hand, there can be also strongly inconsistent defeasible schemas in which inconsistency depends not only on the strict part of the schema, but also on the defeasible part. For example, the schema in figure 4 is inconsistent, since the class A results to have a ranking value of ∞ (the schema declares that the class A is *directly* connected to two incompatible concepts). Now, we can check the results of the defined procedure in the examples presented.

Example 5. Consider example 3. From the translation of the defeasible form of the schema we conclude that the axiom Penguin $\sqsubseteq$ NonFlyingObject has rank 1, while Bird $\sqsubseteq$ WingyObject and Bird $\sqsubseteq$ FlyingObject have rank 0, that means that we end up with two default concepts:

- $\delta_0 := \bigsqcap\{\mathsf{Penguin} \supset \mathsf{NonFlyingObject}, \mathsf{Bird} \supset \mathsf{WingyObject}, \mathsf{Bird} \supset \mathsf{FlyingObject}\}$;
- $\delta_1 := \mathsf{Penguin} \supset \mathsf{NonFlyingObject}$

We can derive the same kind of conclusions as in example 2, and again we can see the limits of the rational closure, since we cannot derive the desirable conclusion that Penguin$\mid\!\sim$WingyObject.

Example 6. Consider the knowledge base obtained in the example 4. We have only a defeasible inclusion axiom Employee $\tilde{\sqsubseteq} \exists \mathsf{f1}^{-}.\mathsf{WorksFor}$, and, since Employee does not turn out to be an exceptional concept, we end up with a single default concept in $\mathcal{B}$:

- $\delta_0 := \{\mathsf{Employee} \supset \exists \mathsf{f1}^{-}.\mathsf{WorksFor}\}$;

Since TopManager is not consistent with all the strict information contained in the schema plus δ_0, we cannot associate δ_0 to TopManager and, despite we have the information that for non-exceptional cases an employee works for a project, we are not forced to conclude that for the exceptional class of the top managers.

6 Conclusions and further work

In this paper we have presented a way to implement a form of defeasible reasoning into the ORM2 formalism. Exploiting the possibility of encoding ORM2$^{\mathsf{zero}}$, that represents a big portion of the ORM2 language, into the description logic $\mathcal{ALCQI}$ on one hand, and a procedure appropriate for modeling one of the main forms of nonmonotonic reasoning, *i.e.* rational closure, into DLs on the other hand, we have defined two new constraints, a *defeasible subclass relation* and a *defeasible mandatory participation*, that are appropriate for modeling defeasible information into ORM2, and that, once translated into $\mathcal{ALCQI}$, allow for the use of the procedures characterizing rational closure to reason about the information contained into an ORM2$^{\mathsf{zero}}$ schema.

The present proposal deals only with reasoning on the information contained in the TBox obtained from an ORM2 schema, but, once we have done the rational closure of

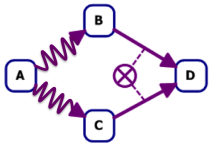

Fig. 4. Inconsistent schema.

the TBox, we can think also of introducing an ABox, that is, the information about a particular domain of individuals. A first proposal in such direction is in [4]. Actually we still lack a complete semantic characterization of rational closure in DLs, but hopefully we shall obtain soon such a result (a first step in such a direction is in [3]). Another future step will be the implementation of nonmonotonic forms of reasoning that extend rational closure, overcoming its inferential limits (see example 2), such as the *lexicographic closure* [16] or the *defeasible inheritance based* approach [5].

References

1. Halpin, T., Morgan, T.: Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design. 2nd edn. Morgan Kaufmann (2001)
2. Bonatti, P.A., Faella, M., Sauro, L.: Defeasible inclusions in low-complexity dls. J. Artif. Intell. Res. (JAIR) **42** (2011) 719–764
3. Britz, K., Meyer, T., Varzinczak, I.: Semantic foundation for preferential description logics. In Wang, D., Reynolds, M., eds.: Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence. Number 7106 in LNAI, Springer (2011) 491–500
4. Casini, G., Straccia, U.: Rational closure for defeasible description logics. In Janhunen, T., Niemelä, I., eds.: JELIA. Volume 6341 of Lecture Notes in Computer Science., Springer (2010) 77–90
5. Casini, G., Straccia, U.: Defeasible inheritance-based description logics. In: IJCAI-11. (2011) 813–818
6. Giordano, L., Olivetti, N., Gliozzi, V., Pozzato, G.L.: Alc + t: a preferential extension of description logics. Fundam. Inform. **96**(3) (2009) 341–372
7. Grimm, S., Hitzler, P.: A preferential tableaux calculus for circumscriptive $\mathcal{ALCO}$. In: RR-09. Number 5837 in LNCS, Berlin, Heidelberg, Springer-Verlag (2009) 40–54
8. Straccia, U.: Default inheritance reasoning in hybrid kl-one-style logics. IJCAI-93 (1993) 676–681
9. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artif. Intell. **55**(1) (1992) 1–60
10. Franconi, E., Mosca, A., Solomakhin, D.: ORM2: formalisation and encoding in OWL2. In: OTM 2012 Workshops. Volume 7567 of LNCS., Springer (2012) 368–378
11. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York, NY, USA (2003)
12. Franconi, E., Mosca, A.: The formalisation of ORM2 and its encoding in OWL2. Technical Report KRDB12-2, KRDB Research Centre, Free University of Bozen-Bolzano (2012) Available at `http://www.inf.unibz.it/krdb/pub/TR/KRDB12-2.pdf`.
13. Calvanese, D., De Giacomo, G., Lenzerini, M.: Identification constraints and functional dependencies in description logics. In: Proceedings of the 17th international joint conference on Artificial intelligence (IJCAI). (2001) 155–160
14. Berardi, D., Cali, A., Calvanese, D., Giacomo, G.D.: Reasoning on UML class diagrams. Art. Intell. **168** (2003)
15. Artale, A., Calvanese, D., Kontchakov, R., Ryzhikov, V., Zakharyaschev, M.: Reasoning over extended ER models. In: Proc. of ER 2007, 26th International Conference on Conceptual Modeling, Springer (2007) 277–292
16. Lehmann, D.J.: Another perspective on default reasoning. Ann. Math. Artif. Intell. **15**(1) (1995) 61–82

OntoMerge: A System for Merging DL-Lite Ontologies

Zhe Wang[1], Kewen Wang[2], Yifan Jin[2], and Guilin Qi[3,4]

[1] University of Oxford, United Kingdom
[2] Griffith University, Australia
[3] Southeast University, China
[4] State Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, China

Abstract. Merging multi-sourced ontologies in a consistent manner is an important and challenging research topic. In this paper, we propose a novel approach for merging DL-Lite$_{bool}^{\mathcal{N}}$ ontologies by adapting the classical model-based belief merging approach, where the minimality of changes is realised via a semantic notion, *model distance*. Instead of using classical DL models, which may be infinite structures in general, we define our merging operator based on a new semantic characterisation for DL-Lite. We show that subclass relation *w.r.t.* the result of merging can be checked efficiently via a QBF reduction. We present our system OntoMerge, which effectively answers subclass queries on the resulting ontology of merging, without first computing the merging results. Our system can be used for answering subclass queries on multiple ontologies.

1 Introduction

Ontologies are widely used for sharing and reasoning over domain knowledge, and their underlying formalisms are often description logics (DLs). To effectively answer queries, ontologies from heterogeneous sources and contributed by various authors are often needed. However, ontologies developed by multiple authors under different settings may contain overlapping, conflicting and incoherent domain knowledge. The ultimate goal of ontology merging is to obtain a single consistent ontology that preserves as much knowledge as possible from two or more heterogeneous ontologies. This is in contrast to ontology matching [5], whose goal is to align entities (with different name) between ontologies, and which is often a pre-stage of ontology merging.

Existing merging systems often adopt formula-based approaches to deal with logical inconsistencies [10; 9; 14]. Most of such approaches can be described as follows: the system first combine the ontologies by taking their union; then, if any inconsistency is detected (through a standard reasoning), it pinpoints the axioms which (may) cause inconsistency; and finally, remove certain axioms to retain consistency. However, such an approach is sometimes unsatisfactory because it is not fine-grained either in the way it measures the minimality of changes, and thus it is often unclear how close the result of merging is to the source ontologies

semantically; or in the way it resolve inconsistency. In [12], an attempt is made to provide some semantic justification for the minimality of changes, however, the result of merging is still syntax-dependant and is often a set of ontologies.

On the other hand, model-based merging operators have been intensively studied in propositional logic, which are syntax-independent and usually satisfy more rationality postulates than formula-based ones. However, a major challenge in adapting model-based merging techniques to DLs is that DL models are generally infinite structures and the number of models of a DL ontology is infinite. Several notions of model distance are defined on classical DL models for ontology revision [13]. Mathematically, it is possible to define a distance o classical DL models. Such a distance is computationally limited as it is unclear how to develop an algorithm for the resulting merging operator. A desirable solution is to define ontology merging operators based on a suitable finite semantic characterisation instead of classical DL models.

In this paper, we focus on merging ontologies expressed as DL-Lite TBoxes, which can be also accompanied with the ontology-based data access (OBDA) framework for data integration [2]. We propose a novel approach for merging ontologies by adapting a classical model-based belief merging approach, where the minimality of changes is realised via a semantic notion, model distance. Instead of using classical DL models, which may be infinite structures in general, we define our merging operator based on the notion of *types*. We show that subclass relation *w.r.t.* the result of merging can be checked efficiently via a QBF reduction, which allows us to make use of the off-the-shelf QBF solvers [8]. We present our system OntoMerge, which effectively answers subclass queries on merging results, without first computing the merging results. Our system can be used to answer subclass queries on multiple ontologies.

2 A New Semantic Characterisation

In our approach, it is sufficient to consider a finite yet large enough signature. A *signature* $\mathcal{S}$ is a union of four disjoint finite sets $\mathcal{S}_C$, $\mathcal{S}_R$, $\mathcal{S}_I$ and $\mathcal{S}_N$, where $\mathcal{S}_C$ is the set of atomic concepts, $\mathcal{S}_R$ is the set of atomic roles, $\mathcal{S}_I$ is the set of individual names and $\mathcal{S}_N$ is the set of natural numbers in $\mathcal{S}$. We assume 1 is always in $\mathcal{S}_N$.

Formally, given a signature $\mathcal{S}$, a DL-Lite$^{\mathcal{N}}_{bool}$ language has the following syntax [1]:

$$R \leftarrow P \mid P^- \qquad S \leftarrow P \mid \neg P$$
$$B \leftarrow \top \mid A \mid \geqslant n\ R \qquad C \leftarrow B \mid \neg C \mid C_1 \sqcap C_2$$

where $n \in \mathcal{S}_N$, $A \in \mathcal{S}_C$ and $P \in \mathcal{S}_R$. B is called a *basic concept* and C is called a *general concept*. $\mathcal{B}_\mathcal{S}$ denotes the set of basic concepts on $\mathcal{S}$. We write $\bot$ for $\neg\top$, $\exists R$ for $\geq 1\ R$, and $C_1 \sqcup C_2$ for $\neg(\neg C_1 \sqcap \neg C_2)$. Let $R^+ = P$, where $P \in \mathcal{S}_R$, whenever $R = P$ or $R = P^-$. A *TBox* $\mathcal{T}$ is a finite set of concept *axioms* of the form $C_1 \sqsubseteq C_2$, where C_1 and C_2 are general concepts. An *ABox* $\mathcal{A}$ is a finite set of membership *assertions* of the form $C(a)$ or $S(a, b)$, where a, b are individual names. In this paper, an ontology is represented as a DL TBox.

The classical DL semantics are given by models. A TBox $\mathcal{T}$ is *consistent* with an ABox $\mathcal{A}$ if $\mathcal{T} \cup \mathcal{A}$ has at least one model. A concept or role is *satisfiable* in $\mathcal{T}$ if it has a non-empty interpretation in some model of $\mathcal{T}$. A TBox $\mathcal{T}$ is *coherent* if all atomic concepts and atomic roles in $\mathcal{T}$ are satisfiable. Note that a coherent TBox must be consistent. TBox $\mathcal{T}$ *entails* an axiom $C \sqsubseteq D$, written $\mathcal{T} \models C \sqsubseteq D$, if all models of $\mathcal{T}$ satisfy $C \sqsubseteq D$. Two TBoxes $\mathcal{T}_1, \mathcal{T}_2$ are *equivalent*, written $\mathcal{T}_1 \equiv \mathcal{T}_2$, if they have the same models.

Now, we introduce a semantic characterisation for DL-Lite TBoxes in terms of *types*. A *type* $\tau \subseteq \mathcal{B}_\mathcal{S}$ is a set of basic concepts over $\mathcal{S}$, such that $\top \in \tau$, and $\geqslant n\ R \in \tau$ implies $\geqslant m\ R \in \tau$ for each pair $m, n \in \mathcal{S}_N$ with $m < n$ and each (inverse) role $R \in \mathcal{S}_R \cup \{\, P^- \mid P \in \mathcal{S}_R \,\}$. Type τ *satisfies* basic concept B if $B \in \tau$, $\neg C$ if τ does not satisfy C, and $C_1 \sqcap C_2$ if τ satisfies both C_1 and C_2. Given a TBox $\mathcal{T}$, type τ satisfies $\mathcal{T}$ if τ satisfies concept $\neg C_1 \sqcup C_2$ for each axiom $C_1 \sqsubseteq C_2$ in $\mathcal{T}$.

For a TBox $\mathcal{T}$, define $\mathsf{TM}(\mathcal{T})$ to be the maximal set of types satisfying the following conditions: (1) all the types in $\mathsf{TM}(\mathcal{T})$ satisfy $\mathcal{T}$; (2) for each type $\tau \in \mathsf{TM}(\mathcal{T})$ and each $\exists R$ in τ, there exists a type $\tau' \in \mathsf{TM}(\mathcal{T})$ (possibly $\tau' = \tau$) containing $\exists R^-$. A type τ is called a *type model* (T-model) of $\mathcal{T}$ if $\tau \in \mathsf{TM}(\mathcal{T})$. Note that $\mathsf{TM}(\mathcal{T})$ is uniquely defined for each TBox $\mathcal{T}$. Note that for a coherent TBox $\mathcal{T}$, $\mathsf{TM}(\mathcal{T})$ is exactly the set of all types satisfying $\mathcal{T}$. Let $\mathbf{TM}(\mathit{\Pi}) = \mathsf{TM}(\mathcal{T}_1) \times \cdots \times \mathsf{TM}(\mathcal{T}_n)$ for $\mathit{\Pi} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$.

Proposition 1. *Given a TBox $\mathcal{T}$, we have the following results:*

- *$\mathcal{T}$ is consistent iff $\mathsf{TM}(\mathcal{T}) \neq \emptyset$.*
- *For a general concept C, C is satisfiable wrt $\mathcal{T}$ iff there exists a T-model in $\mathsf{TM}(\mathcal{T})$ satisfying C.*
- *For two general concepts C, D, $\mathcal{T} \models C \sqsubseteq D$ iff either $\mathsf{TM}(\mathcal{T}) = \emptyset$ or all T-models in $\mathsf{TM}(\mathcal{T})$ satisfy $C \sqsubseteq D$.*
- *$\mathcal{T} \equiv \mathcal{T}'$ iff $\mathsf{TM}(\mathcal{T}) = \mathsf{TM}(\mathcal{T}')$, for any TBox $\mathcal{T}'$.*

Given a type τ, an individual a and an ABox $\mathcal{A}$, we say τ is a *type of a w.r.t. $\mathcal{A}$* if there is a model $\mathcal{I}$ of $\mathcal{A}$ such that $\tau = \{B \mid a^\mathcal{I} \in B^\mathcal{I}, B \in \mathcal{B}_\mathcal{S}\}$. For example, given $\mathcal{A} = \{A(a), \neg B(b), C(c)\}$, type $\tau = \{A, B\}$ is a type of a, but not a type of either b or c in $\mathcal{A}$. For convenience, we will say a *type of a* when the ABox $\mathcal{A}$ is clear from the context. Let $\mathsf{TM}_a(\mathcal{A})$ be the set of all the types of a in $\mathcal{A}$ if a occurs in $\mathcal{A}$; and otherwise, $\mathsf{TM}_a(\mathcal{A})$ be the set of all the types. A set M of T-models *satisfies* an ABox $\mathcal{A}$ if there is a type of a in M, *i.e.*, $M \cap \mathsf{TM}_a(\mathcal{A}) \neq \emptyset$, for each individual a in $\mathcal{A}$.

Proposition 2. *Given a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, $\mathcal{T} \cup \mathcal{A}$ is consistent iff $\mathsf{TM}(\mathcal{T}) \cap \mathsf{TM}_a(\mathcal{A}) \neq \emptyset$ for each a in $\mathcal{A}$.*

3 Merging Operator

In this section, we introduce an approach to merging DL-Lite ontologies to obtain a coherent unified ontology.

An ontology *profile* is of the form $\mathit{\Pi} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$, where $\mathcal{T}_i$ is the ontology from the source n.o. i ($1 \leq i \leq n$). There are two standard definitions of integrity constraints (ICs) in the classical belief change literature [3], the consistency- and entailment-based definitions. We also allow two types of ICs for merging, namely the *consistency constraint* (CC), expressed as a set $\mathcal{A}_c$ of data, and the *entailment constraint* (EC), expressed as a TBox $\mathcal{T}_e$. We assume the IC is self-consistent, that is, $\mathcal{T}_e \cup \mathcal{A}_c$ is always consistent. For an ontology profile $\mathit{\Pi}$, a CC $\mathcal{A}_c$ and a EC $\mathcal{T}_e$, an ontology *merging* operator is a mapping $(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}_c) \mapsto \nabla(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}_c)$, where $\nabla(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}_c)$ is a TBox, *s.t.* $\nabla(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}_c) \cup \mathcal{A}_c$ is consistent, and $\nabla(\mathit{\Pi}, \mathcal{T}_e, \mathcal{A}_c) \models \mathcal{T}_e$.

In classical model-based merging approaches, merging operators are often defined by certain notions of *model distances* [11; 6]. We use $S \triangle S'$ to denote the symmetric difference between two sets S and S', *i.e.*, $S \triangle S' = (S \setminus S') \cup (S' \setminus S)$. Given a set S and a tuple $\boldsymbol{S} = \langle S_1, \ldots, S_n \rangle$ of sets, the *distance* between S and $\boldsymbol{S}$ is defined to be a tuple $\boldsymbol{d}(S, \boldsymbol{S}) = \langle S \triangle S_1, \ldots, S \triangle S_n \rangle$. For two n-element distances $\boldsymbol{d}$ and $\boldsymbol{d'}$, $\boldsymbol{d} \preceq \boldsymbol{d'}$ if $d_i \subseteq d'_i$ for each $1 \leq i \leq n$, where d_i is the i-th element in $\boldsymbol{d}$. Given two sets S and S', define $\sigma(S, S') = S$ if S' is empty, and otherwise, $\sigma(S, S') = \{\, e_0 \in S \mid \exists e'_0 \in S' \text{ } s.t. \text{ } \forall e \in S, \forall e' \in S', d(e, e') \not\prec d(e_0, e'_0) \,\}$. In [6], given a collection $\mathit{\Psi} = \{\varphi_1, \ldots, \varphi_n\}$ of propositional formulas, and some ECs expressed as a propositional theory μ, the result of merging $\varphi_1, \ldots, \varphi_n$ *w.r.t.* μ is the theory whose models are exactly $\sigma(\mathsf{mod}(\mu), \mathsf{mod}(\mathit{\Psi}))$, *i.e.*, those models satisfying μ and having minimal distance to $\mathit{\Psi}$.

Inspired by classical model-based merging, we introduce a merging operator in terms of T-models. For an ontology profile $\mathit{\Pi}$ and an EC $\mathcal{T}_e$, we could define the T-models of the merging to be a subset of $\mathsf{TM}(\mathcal{T}_e)$ (so that $\mathcal{T}_e$ is entailed) consisting of those T-models which have minimal distance to $\mathit{\Pi}$, *i.e.*, $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\mathit{\Pi}))$. However, this straightforward adoption does not take the CC into consideration, and the merging result obtained in this way may not be coherent. For example, let $\mathcal{T}_1 = \{A \sqsubseteq \neg B\}$, $\mathcal{T}_2 = \{\top \sqsubseteq B\}$, $\mathcal{T}_e = \emptyset$, and $\mathcal{A}_c = \{A(a), B(a)\}$. Then, $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\langle \mathcal{T}_1, \mathcal{T}_2 \rangle))$ consists of only one type $\{B\}$. Clearly, the corresponding TBox $\{A \sqsubseteq \bot, \top \sqsubseteq B\}$ does not satisfy the CC, and it is not coherent.

Note that in the above example, once the merging result satisfies the CC, then it is also coherent, because both concepts A and B are satisfiable. In general, it is also the case that coherency can be achieved by applying certain CC to merging. We introduce an auxiliary ABox $\mathcal{A}^\dagger$ in addition to the initial CC $\mathcal{A}_c$, in which each concept and each role is explicitly asserted with a member. That is, $\mathcal{A}^\dagger = \{A(a) \mid A \in \mathcal{S}_C, a \in \mathcal{S}_I \text{ is a fresh individual for } A\} \cup \{P(b, c) \mid P \in \mathcal{S}_R, b, c \in \mathcal{S}_I \text{ are fresh individuals for } P\}$. As assumed, $\mathcal{S}_I$ is large enough for us to take these auxiliary individuals. From the definition of CCs, the merged TBox $\mathcal{T}$ must be consistent with all the assertions in $\mathcal{A}^\dagger$, which assures all the concepts and roles in $\mathcal{T}$ to be satisfiable. Based on this observation, we have the following lemma.

Lemma 1. *$\mathcal{T}$ is coherent iff $\mathcal{T} \cup \mathcal{A}^\dagger$ is consistent for any TBox $\mathcal{T}$.*

To ensure the coherence of merging, we only need to include $\mathcal{A}^\dagger$ into the CC.

For the merging to be consistent with the CC $\mathcal{A}_c$, from Proposition 2, the T-model set M of the merging needs to satisfy $\mathcal{A}_c$. That is, M needs to contain a type of a for each individual a in $\mathcal{A}_c$. However, $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi))$ does not necessarily satisfy this condition, as can be seen from the above example: $\mathsf{TM}_a(\mathcal{A}_c)$ consists of a single type $\{A, B\}$ and $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi)) \cap \mathsf{TM}_a(\mathcal{A}_c) = \emptyset$. Intuitively, for the merging to satisfy the CC, type $\{A, B\}$ need to be added to the T-models of merging. In general, the T-models of merging can be obtained by extending (if necessary) the set $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi))$ with at least one type of a *w.r.t.* $\mathcal{A}_c$ for each individual a in $\mathcal{A}_c$, and if there are multiple such types, choose those with minimal distances.

Based on the above intuitions, the definition of TBox merging is presented as follows.

Definition 1. *Let Π be an ontology profile, $\mathcal{T}_e$ be a TBox, and $\mathcal{A}_c$ be an ABox. Denote $\mathcal{A}^* = \mathcal{A}_c \cup \mathcal{A}^\dagger$. The result of merging Π w.r.t. the EC $\mathcal{T}_e$ and the CC $\mathcal{A}_c$, denoted $\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c)$, is defined as follows*

$$\mathsf{TM}(\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c)) = \sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi)) \cup \bigcup_{a \text{ occurs in } \mathcal{A}^*} \sigma(\mathsf{TM}(\mathcal{T}_e) \cap \mathsf{TM}_a(\mathcal{A}^*), \mathbf{TM}(\Pi)).$$

From the definition, the T-models of the merging are constituted with two parts. The first part consists of those T-models of $\mathcal{T}_e$ (for the satisfaction of the EC) with minimal distances to Π. The second part consists of types of a, for each individual a in $\mathcal{A}^*$, which are added to the first part for the satisfaction of the CC. These types are also required to be T-models of $\mathcal{T}_e$ and have minimal distances to Π. It is clear from Proposition 1 that the result of merging is unique up to TBox equivalence.

4 QBF Reduction

In this section, we consider a standard reasoning problem for ontology merging, namely the *subclass queries*: whether or not the result of merging entails a subclass relation $C \sqsubseteq D$. We present a QBF reduction for this problem, which allows us to make use of the off-the-shelf QBF solvers [8]. We assume that every TBox in the ontology profile is coherent, and in this case, the T-models of a TBox $\mathcal{T}$ are exactly those satisfying $\mathcal{T}$.

We achieve the reduction in three steps. Firstly, we introduce a novel propositional transformation for DL-Lite TBoxes. The transformation is inspired by [1], which contains a transformation from a DL-Lite TBox into a theory in the one variable fragment of first order logic. Considering T-models instead of classical DL models allows us to obtain a simpler transformation to propositional logic than theirs to first order logic.

Function $\phi(\cdot)$ maps a basic concept to a propositional variable, and a general concept (resp., a TBox axiom) to a propositional formula.

$$\begin{aligned}
&\phi(\bot) = \bot, \quad \phi(A) = p_A, \quad \phi(\geqslant n\ R) = p_{nR},\\
&\phi(\neg C) = \neg\phi(C), \qquad \phi(C_1 \sqcap C_2) = \phi(C_1) \wedge \phi(C_2),\\
&\phi(C_1 \sqsubseteq C_2) = \phi(C_1) \rightarrow \phi(C_2).
\end{aligned}$$

Here, p_A and p_{nR} are propositional variables. We use $V_{\mathcal{S}}$ to denote the set of propositional variables corresponding to the basic concepts over $\mathcal{S}$, and we omit the subscript $\mathcal{S}$ in what follows for simplicity. We can see that $\phi(\cdot)$ is a bijective mapping between the set of DL-Lite$^{\mathcal{N}}_{bool}$ general concepts and the set of propositional formulas only referring to symbols in $V_{\mathcal{S}}$ and boolean operators $\neg$, $\wedge$ and $\rightarrow$.

Naturally, given the mapping $\phi(\cdot)$, an arbitrary propositional model may not correspond to a type. We define a formula η whose models are exactly the set of types. Let

$$\eta = \bigwedge_{R^+ \in \mathcal{S}_R} \ \bigwedge_{\substack{m,n \in \mathcal{S}_N \text{ with } m<n \\ \text{and } m<k<n \text{ for no } k \in \mathcal{S}_N}} p_{nR} \rightarrow p_{mR}.$$

Then, $\mathsf{mod}(\eta) = \{\phi(\tau) \mid \tau \text{ is a type}\}$ where $\phi(S)$ stands for for $\{\phi(B_1) \mid B \in S\}$ for a set S of basic concepts.

Given a coherent DL-Lite TBox $\mathcal{T}$, let $\phi(\mathcal{T}) = \bigwedge_{\alpha \in \mathcal{T}} \phi(\alpha) \wedge \eta$. The models of $\phi(\mathcal{T})$ correspond to the T-models of $\mathcal{T}$. For a DL-Lite ABox $\mathcal{A}$ and an individual name a in $\mathcal{A}$, let

$$\phi_a(\mathcal{A}) = \bigwedge_{C(a) \in \mathcal{A}} \phi(C) \wedge \bigwedge_{P \in \mathcal{S}_R} (p_{uP} \wedge p_{vP^-})$$

where u and v are respectively, the maximal number in $\mathcal{S}_N$ *s.t.* $u \leq |\{b_i \mid P(a, b_i) \in \mathcal{A}\}|$ and $v \leq |\{b_i \mid P(b_i, a) \in \mathcal{A}\}|$. Note that we are not transforming an ABox into a propositional theory, but using the encodings $\phi_a(\mathcal{A})$ as constraints over the models.

It is worth noting that the sizes of $\phi(\mathcal{T})$ and $\phi(a, \mathcal{A})$ are both polynomial in the size of $\mathcal{T} \cup \mathcal{A}$. The intuition behind $\phi(\mathcal{T})$ and $\phi(a, \mathcal{A})$ can be shown by the following lemma.

Lemma 2. *Given a coherent TBox $\mathcal{T}$ and an ABox $\mathcal{A}$, then,*

1. $\mathsf{mod}(\phi(\mathcal{T})) = \{\phi(\tau) \mid \tau \in \mathsf{TM}(\mathcal{T})\}$;
2. $\mathsf{mod}(\phi_a(\mathcal{A}) \wedge \eta) = \{\phi(\tau) \mid \tau \in \mathsf{TM}_a(\mathcal{A})\}$.

This transformation essentially allows us to build a connection between our merging operator and propositional belief merging.

Secondly, as we have a transformation from T-models to propositional models, we can encode (minimal) distances between them using QBFs, by extending the encoding in [4], which was introduced for a different purpose. In particular, we need to encode the distances between the models of ϕ and the models of

$\varphi_1, \ldots, \varphi_n$ ($n \geq 1$), where ϕ and φ_i's are propositional formulas in signature V. We make n-fresh copies of V to, informally, encode the models of $\varphi_1, \ldots, \varphi_n$, respectively; let $V^i = \{p^i \mid p \in V\}$ for $1 \leq i \leq n$ where each p^i is a fresh variable for p, and $V^N = \bigcup_{1 \leq i \leq n} V^i$. For a propositional formula φ and $1 \leq i \leq n$, φ^i denotes the formula obtained from φ by replacing each occurrence of p with p^i. We also need another n-fresh copies of V, $V_d^i = \{p_d^i \mid p \in V\}$ to represent the distances. An assignment to V_d^i is expected to capture the difference between a model of ϕ and a model of φ_i, and in particular, p_d^i is assigned true if the truth values of p and p^i are different. Let $V_d^N = \bigcup_{1 \leq i \leq n} V_d^i$. Define

$$F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) = \phi \wedge \bigwedge_{1 \leq i \leq n} \Big(\varphi_i^i \wedge \bigwedge_{p \in V} \big((p \leftrightarrow \neg p^i) \rightarrow p_d^i \big) \Big).$$

A model M of $F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ consists of the assignments to three sets of variables V, V^N and V_d^N. For a set $S \subseteq V \cup V^N \cup V_d^N$, $m(S)$ is the set obtained from S by eliminating the super- and subscripts. Then, $M \cap V$ is a model of ϕ, and $m(M \cap V^i)$ is a model of φ_i. From $(p \leftrightarrow \neg p^i) \rightarrow p_d^i$, $m(M \cap V_d^i)$ *subsumes* the symmetric difference between the former two models. We use $\rightarrow$ instead of $\leftrightarrow$ here, as we will further constraint the assignments of V_d^N to minimal distances.

Furthermore, define a QBF

$$\begin{aligned} D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) = & (\exists V \exists V^N F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)) \wedge \\ & \bigwedge_{\substack{p \in V \\ 1 \leq i \leq n}} \Big(p_d^i \rightarrow \neg \exists V \exists V^N \big(F(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) \wedge (p \leftrightarrow p^i) \big) \Big), \end{aligned}$$

where $\exists V$ with $V = \{p_1, \ldots, p_k\}$ is an abbreviation for $\exists p_1 \ldots \exists p_k$. A model M_d of $D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ is an assignment to V_d^N representing a minimal distance between the models of ϕ and the model tuples of $\langle \varphi_1, \ldots, \varphi_n \rangle$. The first conjunct of $D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ says that there is a model of ϕ and a model tuple of $\langle \varphi_1, \ldots, \varphi_n \rangle$ such that the distance between them is subsumed by $m(M_d)$. The second conjunct checks that $m(M_d)$ is minimal, *i.e.*, there is no distance properly subsumed by $m(M_d)$.

Lemma 3. *Given propositional formulas ϕ and $\varphi_1, \ldots, \varphi_n$, let $MD(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)$ be the set of minimal distances (w.r.t. $\preceq$) between ϕ and $\langle \varphi_1, \ldots, \varphi_n \rangle$ (of the form $\langle M \triangle M_1, \ldots, M \triangle M_n \rangle$ with $M \in \mathsf{mod}(\phi)$ and $M_i \in \mathsf{mod}(\varphi_i)$). Then,*

$$\begin{aligned} \mathsf{mod}(D(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle)) = \{ M_d \subseteq V_d^N \mid & \exists \boldsymbol{d} \in MD(\phi, \langle \varphi_1, \ldots, \varphi_n \rangle) \text{ s.t.} \\ & m(M_d \cap V_d^i) = \boldsymbol{d}_i \text{ for each } 1 \leq i \leq n \}. \end{aligned}$$

Finally, with the encoding of minimal distances, we can encode the T-models of the merging and we are ready to encode the entailment relation. Given an ontology profile $\mathit{\Pi} = \langle \mathcal{T}_1, \ldots, \mathcal{T}_n \rangle$ and a TBox $\mathcal{T}_e$, all the types in $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\mathit{\Pi}))$ satisfy TBox axiom α if and only if QBF $\neg \exists V_d^N E(\mathit{\Pi}, \mathcal{T}_e, \alpha)$ evaluates to true,

with

$$E(\Pi, \mathcal{T}_e, \alpha) = D(\phi(\mathcal{T}_e), \langle \phi(\mathcal{T}_1), \ldots, \phi(\mathcal{T}_n)\rangle) \wedge \\ \neg\forall V\Big(\big(\exists V^N F(\phi(\mathcal{T}_e), \langle \phi(\mathcal{T}_1), \ldots, \phi(\mathcal{T}_n)\rangle)\big) \rightarrow \phi(\alpha)\Big).$$

This QBF can be understood as follows. A model M of $E(\Pi, \mathcal{T}_e, \alpha)$ is an assignment to V_d^N, and represents, by the first conjunct of $E(\Pi, \mathcal{T}_e, \alpha)$, a minimal distance between the T-models of $\mathcal{T}_e$ and the T-model tuples of Π. The second conjunct states the non-entailment, that is, not all of the T-models of $\mathcal{T}_e$ having such a distance satisfy α. The QBF as a whole essentially says that there does not exists a minimal distance $\boldsymbol{d}$ such that a type (in $\sigma(\mathsf{TM}(\mathcal{T}_e), \mathbf{TM}(\Pi))$) selected with $\boldsymbol{d}$ fails to satisfy α. Similarly, given an ABox $\mathcal{A}$ and an individual a in $\mathcal{A}$, all the types in $\sigma(\mathsf{TM}(\mathcal{T}_e) \cap \mathsf{TM}_a(\mathcal{A}), \mathbf{TM}(\Pi))$ satisfy α iff QBF $\neg\exists V_d^N E_a(\Pi, \mathcal{T}_e, \mathcal{A}, \alpha)$ evaluates to true, where $E_a(\Pi, \mathcal{T}_e, \mathcal{A}, \alpha)$ is obtained from $E(\Pi, \mathcal{T}_e, \alpha)$ by replacing $\phi(\mathcal{T}_e)$ with $\phi(\mathcal{T}_e) \wedge \phi_a(\mathcal{A})$.

Now, we can reduce the subclass query answering for TBox merging to QBF as follows.

Theorem 1. *Let Π be an ontology profile with coherent source TBoxes, $\mathcal{T}_e$ be a TBox and $\mathcal{A}_c$ be an ABox in DL-Lite$_{bool}^{\mathcal{N}}$. Let $\mathcal{A}^* = \mathcal{A}_c \cup \mathcal{A}^\dagger$. Given a TBox axiom α, we have $\nabla(\Pi, \mathcal{T}_e, \mathcal{A}_c) \models \alpha$ iff the following QBF evaluates to true*

$$\neg\exists V_d^N \big(E(\Pi, \mathcal{T}_e, \alpha) \vee \bigvee_{a \text{ occurs in } \mathcal{A}^*} E_a(\Pi, \mathcal{T}_e, \mathcal{A}^*, \alpha)\big). \quad (1)$$

5 System Architecture

We have implemented the algorithm for answering subclass queries in ontology merging, called *OntoMerge*, and it is publicly available for test at `http://www.ict.griffith.edu.au/~kewen/OntoMerge/`. The ultimate goal of our system OntoMerge is to transform the entailment over merged ontologies to the validity of QBFs as given in Eq.(1). If the input ontologies are $\mathcal{T}_1, \ldots, \mathcal{T}_n$ and the query is α, then the corresponding QBF can be split into two parts: one is the formula $E(\Pi, \mathcal{T}_e, \alpha)$, and the other is a disjunction of formulas of the form $E_a(\Pi, \mathcal{T}_e, \mathcal{A}^*, \alpha)$. The size of the first part is only determined by the sizes of the input ontologies, the input EC and the query α, and the size of α is often small compared to the ontologies. In the second part of the resulting QBF, the number of disjuncts is essentially determined by the number of unsatisfiable concepts in the union of input ontologies (as will be explained later) and the input CC. Thus, the number of unsatisfiable concepts plays a crucial role in the complexity of the reduction algorithm.

In OntoMerge, we first check whether the given ontologies can be simply jointed without causing any incoherency using an off-the-shelf DL reasoner (HermiT [7] is used in the current program). The set of unsatisfiable concepts will be stored for being used later. If the union of input ontologies is coherent, the

query answering can be done by the DL reasoner; otherwise, the system generates the QBF specified in Eq.(1). For this purpose, the system will first scan all input ontologies to obtain the set of basic concepts occurring in these ontologies and assign a propositional variable with each basic concept. Then the QBF is generated in QBF 1.0 format[5]. The structure of OntoMerge is depicted in Figure 1.

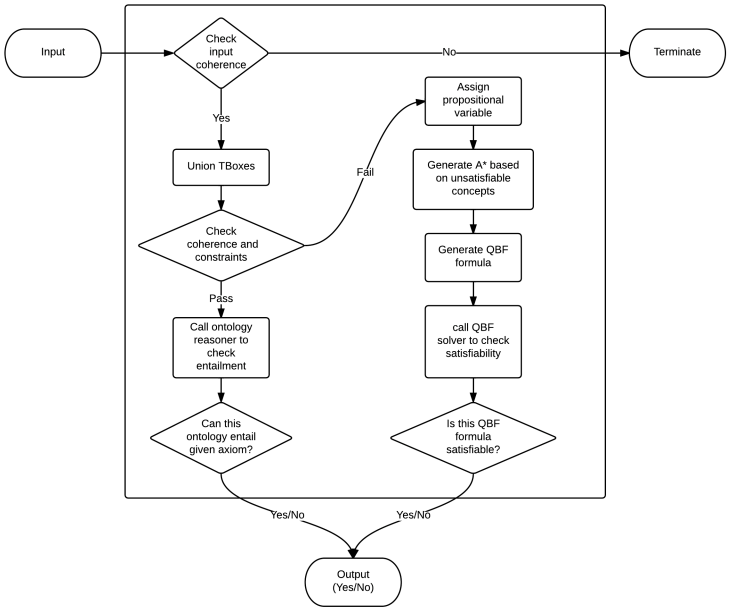

Fig. 1. System Structure of OntoMerge

However, as most of efficient QBF solvers accept only input QBFs in the prenex normal form, the QBF generated in this way cannot be directly fed to a QBF solver. So we need to convert the QBF into the prenex normal form first. Unfortunately, the standard translation from a given QBF into its prenex normal form is very inefficient and thus several heuristics based on the specific QBF are used to optimize the efficiency in our implementation. For instance, from Eq.(1), we can see that the major source for introducing a large number of new variables is from the construction of the ABox $\mathcal{A}^*$. So we introduce new variables for only those concepts that are unsatisfiable and add new assertions for such new variables to $\mathcal{A}^*$ in the reduction algorithm. This optimization significantly reduces the number of new variables.

[5] `http://qbflib.org/boole.html`

Once the QBF is generated, a publicly available program is used to convert it from QBF 1.0 format to QDIMACS format[6] or ISCAS format[7] and then an efficient QBF solver is used to decide the validity of the QBF.

6 Experimental Results

To test the efficiency of OntoMerge, we used the DL-Lite$^{\mathcal{N}}_{bool}$ fragment of the medical ontology Galen[8], which is of medium size. Our experimental results show that the system is relatively efficient, while further optimization is still under way. Specifically, various randomly modified fragments of Galen were merged using OntoMerge and the following three types of experiments were conducted:

- Fixed total number of axioms in the input ontologies but varied number of unsatisfiable concepts.
- Fixed total number of unsatisfiable concepts but varied total number of axioms in the input ontologies.
- Fixed number of unsatisfiable concepts but varied total number of input ontology axioms.

A PC with Intel Core 2 Duo E8400, 4GB RAM, running Linux Mint 13 64bit, and CirQit QBF solver [8] were used in our tests. For each test, the time is limited to one hour (i. e. the program will be terminated after one hour no matter a result is returned or not).

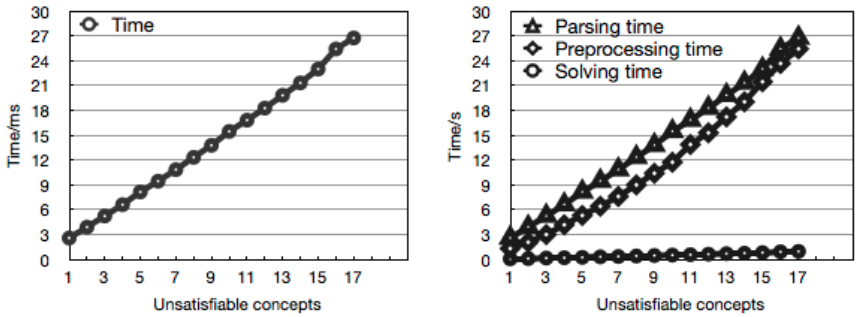

Fig. 2. Total axioms:42, unsatisfiable concepts from 1 to 18

Figure 2 shows the experimental results in the first set of tests. In each test, 42 axioms were randomly selected from Galen ontology and they were separated into two sub-ontologies for merging. Then assertions were inserted into one of them so that some concepts became unsatisfiable in the union of these two ontologies. The number of unsatisfiable concepts varied from 1 to 18. From this

[6] http://www.qbflib.org/qdimacs.html
[7] http://logic.pdmi.ras.ru/~basolver/rtl.html
[8] http://www.co-ode.org/galen/

figure we can see that the program is quite fast according to the time used for generating the QBF and the time used for deciding the validity of the QBF. This is because when the number of unsatisfiable concepts increases, the size of the QBF generated increases linearly. However, when the number of unsatisfiable concepts is over 18, OntoMerge can still generate the QBF but the QBF solver is unable to decide the validity of such a QBF.

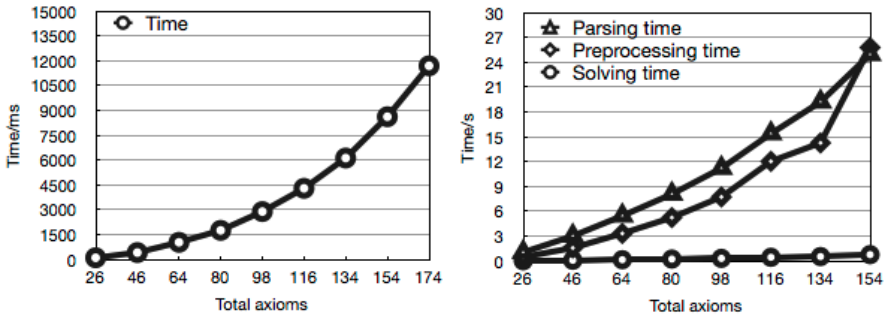

Fig. 3. Total number of unsatisfiable concept:1, total axioms from 26 to 174

In the second set of tests, we fixed the number of unsatisfiable concepts to 1 but increased the total number of axioms from 26 to 174. Figure 3 shows that when the number of axioms is increased, the time cost for generating QBF increases faster than in the first set of tests. This is partly due to the fact that with the increase of the total axioms, the size of the QBF significantly increases too. Similar to the case for the first set of tests, when the total number of axioms is over 174, the QBF solver fails again.

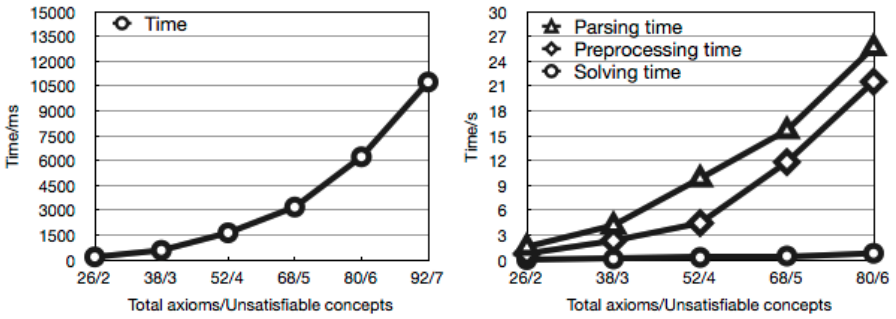

Fig. 4. Ratio of unsatisfiable concepts to total axioms = 0.15

In the third set of tests, we fix the ratio of total number of axioms to the number of unsatisfiable concepts to around 0.15 but let the total number of axioms varied from 26 to 92 as well as the number of unsatisfiable concepts. Figure 4 shows a similar pattern to Figure 3. When the total number of axioms is over 92, the QBF solver failed to return an answer.

7 Conclusion

We have developed a novel approach for merging ontologies in DL-Lite, in terms of types instead of classical DL models. We have also presented algorithms to reduce subclass queries of DL-Lite merging to the evaluation of QBFs and thus provided a novel way of reasoning with the result of ontology merging using efficient QBF solvers. We have implemented a preliminary merging system OntoMerge, and reported some experimental results in the paper. Currently we are extending the approach in two directions: (1) merging DL-Lite ontologies with both TBoxes and ABoxes, and (2) merging ontologies in expressive DLs.

Acknowledgement: We would like to thank the three anonymous referees for their helpful comments. Special thanks to Rodney Topor for various discussions with him. Kewen Wang was partially supported by the ARC grants DP1093652 and DP110101042. Guilin Qi was partially supported by the NSFC grant 61272378.

References

1. A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. The DL-Lite family and relations. *J. Artif. Intell. Res.*, 36:1–69, 2009.
2. D. Calvanese and G. De Giacomo. Data integration: A logic-based perspective. *AI Magazine*, 26(1):59–70, 2005.
3. J. P. Delgrande and T. Schaub. A consistency-based approach for belief change. *Artif. Intell.*, 151(1-2):1–41, 2003.
4. J. P. Delgrande, T. Schaub, H. Tompits, and S. Woltran. On computing belief change operations using quantified boolean formulas. *J. Log. Comput.*, 14(6):801–826, 2004.
5. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, 2007.
6. P. Everaere, S. Konieczny, and P. Marquis. Conflict-based merging operators. In *Proc. KR*, 348–357, 2008.
7. I. Horrocks, B. Motik and Z. Wang. The HermiT OWL Reasoner. *Proc. IJCAR-ORE workshop*, 2012.
8. R. Goultiaeva, V. Iverson, and F. Bacchus. Beyond CNF: A circuitbased QBF solver. In *Proc. SAT*, 412–426, 2009.
9. A. Kalyanpur, B. Parsia, E. Sirin, and B. Cuenca Grau. Repairing unsatisfiable concepts in OWL ontologies. In *Proc. of ESWC*, pages 170–184, 2006.
10. A. Kalyanpur, B. Parsia, E. Sirin, and J. A. Hendler. Debugging unsatisfiable classes in owl ontologies. *J. Web Semantics*, 3(4):268–293, 2005.
11. S. Konieczny and R. Pino Pérez. Merging information under constraints: A logical framework. *J. Log. Comput.*, 12(5):773–808, 2002.
12. T. Meyer, K. Lee, and R. Booth. Knowledge integration for description logics. In *Proc. AAAI*, 645–650, 2005.
13. G. Qi and J. Du. Model-based revision operators for terminologies in description logics. In *Proc. IJCAI*, 891-897, 2009.
14. S. Schlobach, Z. Huang, R. Cornet, and F. van Harmelen. Debugging incoherent terminologies. *J. Autom. Reasoning*, 39(3):317–349, 2007.
15. Z. Wang, K. Wang, and R. Topor. A new approach to knowledge base revision in DL-Lite. In *Proc. AAAI*, 369–374, 2010.

Lexicographic Closure for Defeasible Description Logics

Giovanni Casini[1] and Umberto Straccia[2]

[1] Centre for Artificial Intelligence Research, CSIR Meraka Institute and UKZN, South Africa
Email: GCasini@csir.co.za
[2] Istituto di Scienza e Tecnologie dell'Informazione (ISTI - CNR), Pisa, Italy
Email: straccia@isti.cnr.it

Abstract. In the field of non-monotonic logics, the *lexicographic closure* is acknowledged as a a powerful and logically well-characterized approach; we are going to see that such a construction can be applied in the field of Description Logics, an important knowledge representation formalism, and we shall provide a simple decision procedure.

1 Introduction

The application of non-monotonic reasoning (see, *e.g.* [12]) to Description Logics (DLs) [1] has received a lot of attention in the last years, resulting in the development of various proposals, such as [2, 4, 6, 5, 7–11, 13–15, 21, 22]. However, between them just a few ([8–10, 14, 22]) take under consideration the preferential approach (see [19]), a well-known approach to non-monotonic reasoning. Here we take under consideration one of the main proposals in the preferential area, the *lexicographic closure* [18], developed by Lehmann for propositional languages, and never taken under consideration in the DL field. We are going to readapt such a procedure to $\mathcal{ALC}$, a significant and expressive representative of the various DLs. The procedure we are going to implement is obtained by modifying the rational closure construction defined for $\mathcal{ALC}$ in [9].

We proceed as follows: we present a construction of the lexicographic closure at the propositional level that slightly generalizes the construction presented by Lehmann, and still based on classical entailment tests only; then we implement such a construction in $\mathcal{ALC}$. Due to the space limits, we shall omit the proofs of the presented propositions.

2 Propositional Lexicographic Closure

Let ℓ be a finitely generated classical propositional language, defined in the usual way using $\neg, \wedge, \vee, \rightarrow$ as connectives, $C, D, \ldots$ as sentences, $\Gamma, \Delta, \ldots$ as finite sets of sentences, $\top$ and $\bot$ as abbreviations for, respectively, $A \vee \neg A$ and $A \wedge \neg A$ for some A. The symbols $\models$ and $\mid\!\sim$ will indicate, respectively, the classical consequence relation and a defeasible consequence relation. An element of a consequence relation, $\Gamma \models C$ or $\Gamma \mid\!\sim C$, will be called a *sequent*, and in the case of $\mid\!\sim$ it has to be read as 'If Γ, then typically C'.

We call *conditional knowledge base* a pair $\langle \mathcal{T}, \mathcal{B} \rangle$, where $\mathcal{T}$ is a set of *strict* sequents $C \models D$, representing certain knowledge, and $\mathcal{B}$ is a set of *defeasible* sequents $C \mid\!\sim D$, representing default information.

Example 1. The typical 'penguin' example can be encoded as [3]: $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ with $\mathcal{T} = \{P \models B\}$ and $\mathcal{B} = \{P \mathrel{|\!\sim} \neg F, B \mathrel{|\!\sim} F\}$. □

In what follows we are going to present a slight generalization of Lehmann's procedure [18], that is, a non-monotonic reasoning procedure that relies on a decision procedure for $\models$ only. We then suggest how to transpose such an approach into the framework of DLs.

In the present section we shall proceed in the following way: first, we define the notion of *rational consequence relation* (see *e.g.* [17]) and we present the notions of *rational* and *lexicographic closure*; then, we describe a procedure to build a *lexicographic closure* of a conditional knowledge base.

Rational Consequence Relations. The preferential approach is mainly based the identification of the structural properties that a well-behaved (both from the intuitive and the logical point of view) non-monototonic consequence relation should satisfy. Between the possible interesting options, a particular relevance has been given to the group of properties characterizing the class of *rational consequence relations* (see [17]). A consequence relation $\mathrel{|\!\sim}$ is *rational* iff it satisfies the following properties:

(REF)	$C \mathrel{|\!\sim} C$	Reflexivity			
(CT)	$\dfrac{C \mathrel{|\!\sim} D \quad C \wedge D \mathrel{|\!\sim} F}{C \mathrel{|\!\sim} F}$	Cut (Cumulative Trans.)	(RW)	$\dfrac{C \mathrel{|\!\sim} D \quad D \models F}{C \mathrel{|\!\sim} F}$	Right Weakening
(CM)	$\dfrac{C \mathrel{|\!\sim} D \quad C \mathrel{|\!\sim} F}{C \wedge D \mathrel{|\!\sim} F}$	Cautious Monotony	(OR)	$\dfrac{C \mathrel{|\!\sim} F \quad D \mathrel{|\!\sim} F}{C \vee D \mathrel{|\!\sim} F}$	Left Disjunction
(LLE)	$\dfrac{C \mathrel{|\!\sim} F \quad \models C \leftrightarrow D}{D \mathrel{|\!\sim} F}$	Left Logical Equival.	(RM)	$\dfrac{C \mathrel{|\!\sim} F \quad C \not\mathrel{|\!\sim} \neg D}{C \wedge D \mathrel{|\!\sim} F}$	Rational Monotony

We refer the reader to [19] for an insight of the meaning of such rules. (RM) is generally considered as the strongest form of monotonicity we can use in the characterization of a reasoning system in order to formalise a well-behaved form of defeasible reasoning. The kind of reasoning we want to implement applies at the level of sequents: let $\mathcal{B} = \{C_1 \mathrel{|\!\sim} E_1, \ldots, C_n \mathrel{|\!\sim} E_n\}$ be a conditional knowledge base; we want the agent to be able to reason about the defeasible information at its disposal, that is, to be able to derive new sequents from his conditional base.

Semantically, rational consequence relations can be characterized by means of a particular kind of possible-worlds model, that is, ranked preferential models, but we shall not deepen the connection with such a semantical characterization here (see [17]). Except for (RM), all the above properties are *closure properties*, that is, they are preserved under intersection of the consequence relations, allowing for the definition of a notion of entailment (see [16]), called *preferential closure*; on the other hand, (RM) is not preserved under intersection, and not every rational consequence relation describes a desirable and intuitive form of reasoning. Two main constructions have been proposed to define interesting and well-behaved rational consequence relations: the rational closure in [17], and the lexicographic closure in [18].

Lehmann and Magidor's *rational closure* operation behaves in an intuitive way and it is logically strongly characterized (we refer the reader to [17, 9] for a description of

[3] Read B as 'Bird', P as 'Penguin' and F as 'Flying'.

rational closure). Notwithstanding, rational closure is considered a too weak form of reasoning, since often we cannot derive intuitive conclusions from our premises. The main problem of the rational closure is that if an individual falls under an atypical subclass (for example, *penguins* are an atypical subclass of *birds*, since they do not fly), we cannot associate to it *any* of the typical properties characterizing the superclass.

Example 2. Consider the KB in example 1, and add to the set $\mathcal{B}$ the sequent $B|\sim T$ (read T as 'Has feathers'). Even if it would be desirable to conclude that penguins have feathers ($P|\sim T$), in the rational closure it is not possible, since penguins, being atypical birds, are not allowed to inherit *any* of the typical properties of birds.

Rational closure fails to respect the *presumption of independence* ([18], pp.63-64): even if a class does not satisfy a typical property of a superclass, we should presume that it behaves in a typical way w.r.t. the other properties, if not forced to conclude the contrary. In an attempt to overcome such a shortcoming, Lehmann has proposed in [18] a possible extension of rational closure, in order to preserve the desirable logical properties of rational closure, but augmenting in an intuitive way its inferential power.

Lexicographic Closure. In this paragraph we are going to present the procedure to define the lexicographic closure of a conditional knowledge base. Our procedure just slightly generalizes the one in [18], considering also knowledge bases $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ with a strict part $\mathcal{T}$.

The essence of the procedure to build the lexicographic closure of $\mathcal{K}$ consists in transforming $\langle \mathcal{T}, \mathcal{B} \rangle$ into a KB $\langle \Phi, \Delta \rangle$, where Φ and Δ are sets containing formulae instead of sequents; that is, Φ contains what we are informed to be *necessarily* true, while Δ contains the formulae we consider to be *typically*, but not necessarily true. Once we have defined the pair $\langle \Phi, \Delta \rangle$, we can easily define from it the lexicographic closure of $\mathcal{K}$.

So, consider $\langle \mathcal{T}, \mathcal{B} \rangle$, with $\mathcal{B} = \{C_1|\sim E_1, \ldots, C_n|\sim E_n\}$. The steps for the construction of $\langle \Phi, \Delta \rangle$ (obtained by combining the construction in [18] with some results from [3]) are the following:

Step 1. We transfer the information in $\mathcal{T}$ into correspondent $|\sim$-sequents and add it to $\mathcal{B}$, that is, we move from a characterization $\langle \mathcal{T}, \mathcal{B} \rangle$ to $\langle \emptyset, \mathcal{B}' \rangle$, where $\mathcal{B}' = \mathcal{B} \cup \{(C \wedge \neg D)|\sim \bot \mid C \models D \in \mathcal{T}\}$. Intuitively, $C \models D$ is equivalent to saying that its negation is an absurdity, *i.e.* $(C \wedge \neg D)|\sim \bot$ (see [3], Section 6.5).

Step 2. We define $\Delta_{\mathcal{B}'}$ as the set of the *materializations* of the sequents in $\mathcal{B}'$, *i.e.* the material implications corresponding to such sequents: $\Delta_{\mathcal{B}'} = \{C \rightarrow D \mid C|\sim D \in \mathcal{B}'\}$. Also, we indicate by $\mathfrak{A}_{\mathcal{B}'}$ the set of the antecedents of the sequents in $\mathcal{B}'$: $\mathfrak{A}_{\mathcal{B}'} = \{C \mid C|\sim D \in \mathcal{B}'\}$.

Step 3. Now we define an *exceptionality ranking* of sequents w.r.t. $\mathcal{B}'$.

Exceptionality: Lehmann and Magidor call a formula C *exceptional* for a set of sequents $\mathcal{D}$ iff it is false in all the most typical situations satisfying $\mathcal{D}$ (see [17], Section 2.6); in particular, C is exceptional w.r.t. $\mathcal{D}$ if we can derive $\top|\sim \neg C$ from the preferential closure of $\mathcal{D}$. $C|\sim D$ is said to be exceptional for $\mathcal{D}$ iff its antecedent C is exceptional for $\mathcal{D}$. Exceptionality of sequents can be decided based on $\models$ only (see [17], Corollary 5.22), as C is exceptional for a set of sequents $\mathcal{D}$ iff $\Delta_{\mathcal{D}} \models \neg C$.

Given a set of sequents $\mathcal{D}$, indicate by $E(\mathfrak{A}_{\mathcal{D}})$ the set of the antecedents that result exceptional w.r.t. $\mathcal{D}$, that is $E(\mathfrak{A}_{\mathcal{D}}) = \{C \in \mathfrak{A}_{\mathcal{D}} \mid \Delta_{\mathcal{D}} \models \neg C\}$, and with $E(\mathcal{D})$ the

exceptional sequents in $\mathcal{D}$, *i.e.* $E(\mathcal{D}) = \{C|\sim D \in \mathcal{D} \mid C \in E(\mathfrak{A}_\mathcal{D})\}$. Obviously, for every $\mathcal{D}$, $E(\mathcal{D}) \subseteq \mathcal{D}$.

Step 3.1. We can construct iteratively a sequence $\mathcal{E}_0, \mathcal{E}_1 \ldots$ of subsets of the conditional base $\mathcal{B}'$ in the following way: $\mathcal{E}_0 = \mathcal{B}'$, $\mathcal{E}_{i+1} = E(\mathcal{E}_i)$. Since $\mathcal{B}'$ is a finite set, the construction will terminate with an empty ($\mathcal{E}_n = \emptyset$) or a finite ($\mathcal{E}_n + 1 = \mathcal{E}_n \neq \emptyset$) fixed point of E.

Step 3.2. Using such a sequence, we can define a ranking function r that associates to every sequent in $\mathcal{B}'$ a number, representing its level of exceptionality:

$$r(C|\sim D) = \begin{cases} i & \text{if } C|\sim D \in \mathcal{E}_i \text{ and } C|\sim D \notin \mathcal{E}_{i+1} \\ \infty & \text{if } C|\sim D \in \mathcal{E}_i \text{ for every } i\ . \end{cases}$$

Step 4. In Step 3, we defined the materialization of $\mathcal{B}'$ and the rank of every sequent in it. Now,

Step 4.1. we can determine if $\mathcal{B}'$ is inconsistent. A conditional base is inconsistent if in its preferential closure we obtain the sequent $\top|\sim\bot$ (from this sequent we can derive any other sequent using (RW) and (CM)). Given the result in Step 3.1, we can check the consistency of $\mathcal{B}'$ using $\Delta_{\mathcal{B}'}$: $\top|\sim\bot$ is in the preferential closure of $\mathcal{B}'$ iff $\Delta_{\mathcal{B}'} \models \bot$.

Step 4.2. Assuming $\mathcal{B}'$ is consistent, and given the ranking, we define the *background theory* $\widetilde{\mathcal{T}}$ of the agent as $\widetilde{\mathcal{T}} = \{\top \models \neg C \mid C|\sim D \in \mathcal{B}' \text{ and } r(C|\sim D) = \infty\}$[4] , and a correspondent set of formulae $\Phi = \{\neg C \mid \top \models \neg C \in \widetilde{\mathcal{T}}\}$ (one may verify that, modulo classical logical equivalence, $\mathcal{T} \subseteq \widetilde{\mathcal{T}}$).

Step 4.3. once we have $\widetilde{\mathcal{T}}$, we can also identify the set of sequents $\widetilde{\mathcal{B}}$, *i.e.*, the defeasible part of the information contained in $\mathcal{B}'$: $\widetilde{\mathcal{B}} = \{C|\sim D \in \mathcal{B}' \mid r(C|\sim D) < \infty\}$ (one may verify that $\widetilde{\mathcal{B}} \subseteq \mathcal{B}$). We indicate the ranking of $\widetilde{\mathcal{B}}$ as the highest ranking value of the conditionals in $\widetilde{\mathcal{B}}$, *i.e.* $r(\widetilde{\mathcal{B}}) = \max\{r(C|\sim D) \mid C|\sim D \in \widetilde{\mathcal{B}}\}$.

Essentially, so far we have moved the non-defeasible knowledge 'hidden' in $\mathcal{B}$ to $\mathcal{T}$.

Step 5. Now we can define the lexicographic closure of $\langle\widetilde{\mathcal{T}}, \widetilde{\mathcal{B}}\rangle$. Consider the set of the *materializations* of the sequents in $\widetilde{\mathcal{B}}$, $\Delta = \{C \to D \mid C|\sim D \in \widetilde{\mathcal{B}}\}$, and assume that $r(\widetilde{\mathcal{B}}) = k$. Δ^k represents the subset of Δ composed by conditionals of rank k, *i.e.* $\Delta^i = \{C \to D \in \Delta \mid r(C|\sim D) = i\}$. We can associate to every subset $\mathcal{D}$ of Δ a string of natural numbers $\langle n_0, \ldots, n_k\rangle_\mathcal{D}$, where $n_0 =\mid \mathcal{D} \cap \Delta^k \mid$, and, in general, $n_i =\mid \mathcal{D} \cap \Delta^{k-i} \mid$. In this way we obtain a string of numbers expressing how many materializations of defaults are contained in $\mathcal{D}$ for every ranking value. We can order linearly the tuples $\langle n_0, \ldots, n_k\rangle_\mathcal{D}$ using the classic lexicographic order: $\langle n_0, \ldots, n_k\rangle \geq \langle m_0, \ldots, m_k\rangle$ iff
(i) for every i ($0 \leq i \leq k$), $n_i \geq m_i$, or
(ii) if $n_i < m_i$, there is a j s.t. $j < i$ and $n_j > m_j$.

[4] One may easily verify the correctness of this definition referring to the following results in [3], Section 7.5.3: Definition 7.5.1, the definition of *clash* (p.178), Corollary 7.5.7, Definition 7.5.2, and Lemma 7.5.5. It suffices to show that the set of the sequents with ∞ as ranking value represents the greatest clash of $\mathcal{B}$, which can be proved quite immediately by the definition of the exceptionality ranking.

This lexicographic order is based on the intuitions that the more conditionals are satisfied, the better it is, and that more exceptional conditionals have the priority over less exceptional ones. The linear ordering $>$ between the tuples corresponds to a modular ordering $\prec$ between the subsets of Δ:
Seriousness ordering $\prec$ ([18], Definition 2). Let $\mathcal{D}$ and $\mathcal{E}$ be subsets of Δ. $\mathcal{D}$ is preferred to (*more serious than*) $\mathcal{E}$ ($\mathcal{D} \prec \mathcal{E}$) iff $\langle n_0, \ldots, n_k \rangle_{\mathcal{D}} > \langle n_0, \ldots, n_k \rangle_{\mathcal{E}}$.

Given a conditional knowledge base $\langle \mathcal{T}, \mathcal{B} \rangle$ (transformed into $\langle \Phi, \Delta \rangle$) and a set of premises Γ, we indicate by $\mathfrak{D}$ the set of the preferred subsets of Δ that are consistent with the certain information we have at our disposal, that is Φ and Γ:

$$\mathfrak{D}_\Gamma = \min_{\prec}\{\mathcal{D} \subseteq \Delta \mid \Gamma \cup \Phi \cup \mathcal{D} \not\models \bot\}$$

The consequence relation $\mid\!\sim^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle}$, corresponding to the lexicographic closure of $\langle \mathcal{T}, \mathcal{B} \rangle$, will be defined as:

$$\Gamma \mid\!\sim^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle} E \text{ iff } \Gamma \cup \Phi \cup \mathcal{D} \models E \text{ for every } \mathcal{D} \in \mathfrak{D}_\Gamma.$$

The procedure proposed by Lehmann relies heavily on a proposal by Poole, [20], but it makes also use of the cardinality and the exceptionality ranking of the sets of defaults. At the propositional level, the problem of deciding if a sequent $C \mid\!\sim D$ is in the lexicographic closure of a conditional base $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ is exponential (see [18], p.81).

Example 3. Consider the knowledge base in the example 2: $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$ with $\mathcal{T} = \{P \models B\}$ and $\mathcal{B} = \{P \mid\!\sim \neg F, B \mid\!\sim F, B \mid\!\sim T\}$. We define **(Step 1)** the set $\mathcal{B}' = \{P \wedge \neg B \mid\!\sim \bot, P \mid\!\sim \neg F, B \mid\!\sim F, B \mid\!\sim T\}$, and the ranking values obtained from the materialization of $\mathcal{B}'$ are $r(B \mid\!\sim F) = r(B \mid\!\sim T) = 0$, $r(P \mid\!\sim \neg F) = 1$, $r(P \wedge \neg B \mid\!\sim \bot) = \infty$ **(Steps 2-3)** . Hence, we end up with a pair $\langle \Phi, \Delta \rangle$, with $\Phi = \{\neg(P \wedge \neg B)\}$, $\Delta = \Delta^0 \cup \Delta^1$, $\Delta^0 = \{B \rightarrow F, B \rightarrow T\}$, and $\Delta^1 = \{P \rightarrow \neg F\}$ **(Steps 4-5)**. We want to check if we can derive $P \mid\!\sim T$, impossible to derive from the rational closure of $\mathcal{K}$. We have to find which are the most serious subsets of Δ that are consistent with P and Φ: it turns out that there is only one, $\mathcal{D} = \{B \rightarrow T, P \rightarrow \neg F\}$, and we have $\{P\} \cup \Phi \cup \mathcal{D} \models T$, that is, $P \mid\!\sim^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle} T$.

3 Lexicographic Closure in $\mathcal{ALC}$

Now we redefine Lehmann's procedure for the DL case. We consider a significant DL representative, namely $\mathcal{ALC}$ (see e.g. [1], Chap. 2). $\mathcal{ALC}$ corresponds to a fragment of first order logic, using monadic predicates, called *concepts*, and diadic ones, called *roles*. To ease the reader in taking account of the correspondence between the procedure presented in Section 2 and the proposal in $\mathcal{ALC}$, we are going to use the same notation for the components playing an analogous role in the two constructions: capital letters $C, D, E, \ldots$ now indicate *concepts*, instead of propositions, and $\models$ and $\mid\!\sim$ to indicate, respectively, the 'classical' consequence relation of $\mathcal{ALC}$ and a non-monotonic consequence relation in $\mathcal{ALC}$. We have a finite set of *concept names* $\mathcal{C}$, a finite set of *role names* $\mathcal{R}$ and the set $\mathcal{L}$ of $\mathcal{ALC}$ *-concepts* is defined inductively as follows: (i) $\mathcal{C} \subset \mathcal{L}$; (ii) $\top, \bot \in \mathcal{L}$; (iii) $C, D \in \mathcal{L} \Rightarrow C \sqcap D, C \sqcup D, \neg C \in \mathcal{L}$; and (iii)

$C \in \mathcal{L}, R \in \mathcal{R} \Rightarrow \exists R.C, \forall R.C \in \mathcal{L}$. Concept $C \rightarrow D$ is used as a shortcut of $\neg C \sqcup D$. The symbols $\sqcap$ and $\sqcup$ correspond, respectively, to the conjunction $\wedge$ and the disjunction $\vee$ of classical logic. Given a set of *individuals* $\mathcal{O}$, an *assertion* is of the form $a{:}C$ ($C \in \mathcal{L}$) or of the form $(a,b){:}R$ ($R \in \mathcal{R}$), respectively indicating that the individual a is an instance of concept C, and that the individuals a and b are connected by the role R. A *general inclusion axiom* (GCI) is of the form $C \sqsubseteq D$ ($C, D \in \mathcal{L}$) and indicates that any instance of C is also an instance of D. We use $C = D$ as a shortcut of the pair of $C \sqsubseteq D$ and $D \sqsubseteq C$.

From a FOL point of view, concepts, roles, assertions and GCIs, may be seen as formulae obtained by the following transformation

$$\begin{array}{ll} \tau(a{:}C) & = \tau(a, C) \\ \tau((a,b){:}R) & = R(a,b) \\ \tau(C \sqsubseteq D) & = \forall x.\tau(x,C) \rightarrow \tau(x,D) \\ \tau(x, A) & = A(x) \\ \tau(x, \neg C) & = \neg\tau(x, C) \end{array} \qquad \begin{array}{ll} \tau(x, C \sqcap D) & = \tau(x,C) \wedge \tau(x,D) \\ \tau(x, C \sqcup D) & = \tau(x,C) \vee \tau(x,D) \\ \tau(x, \exists R.C) & = \exists y.R(x,y) \wedge \tau(y,C) \\ \tau(x, \forall R.C) & = \forall y.R(x,y) \rightarrow \tau(y,C)\,. \end{array}$$

Now, a classical knowledge base is defined by a pair $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{T}$ is a finite set of GCIs (a *TBox*) and $\mathcal{A}$ is a finite set of assertions (the *ABox*), whereas a *defeasible knowledge base* is represented by a triple $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{B} \rangle$, where additionally $\mathcal{B}$ is a finite set of *defeasible inclusion axioms* of the form $C \mathrel{|\!\sim} D$ ('an instance of a concept C is typically an instance of a concept D'), with $C, D \in \mathcal{L}$.

Example 4. Consider Example 3. Just add a role $Prey$ in the vocabulary, where a role instantiation $(a,b){:}Prey$ is read as 'a preys on b', and add also two more concepts, I (Insect) and Fi (Fish). A defeasible KB is $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{B} \rangle$ with $\mathcal{A} = \{a{:}P, b{:}B, (a,c){:}Prey, (b,c){:}Prey\}$; $\mathcal{T} = \{P \sqsubseteq B, I \sqsubseteq \neg Fi\}$ and $\mathcal{B} = \{P \mathrel{|\!\sim} \neg F,\ B \mathrel{|\!\sim} F,\ B \mathrel{|\!\sim} T,\ P \mathrel{|\!\sim} \forall Prey.Fi,\ B \mathrel{|\!\sim} \forall Prey.I\}$. □

The particular structure of a defeasible KB allows for the 'isolation' of the pair $\langle \mathcal{T}, \mathcal{B} \rangle$, that we could call the *conceptual system* of the agent, from the information about the individuals (formalised in $\mathcal{A}$) that will play the role of the facts known to be true. In the next section we are going to work with the information about concepts $\langle \mathcal{T}, \mathcal{B} \rangle$ first, exploiting the immediate analogy with the homonymous pair of Section 2, then we will address the case involving individuals as well.

Construction of the Lexicographic Closure. We apply to $\langle \mathcal{T}, \mathcal{B} \rangle$ a procedure analogous to the propositional one, in order to obtain from $\langle \mathcal{T}, \mathcal{B} \rangle$ a pair $\langle \Phi, \Delta \rangle$, where Φ and Δ are two sets of concepts, the former representing the background knowledge, that is, concepts necessarily applying to each individual, the latter playing the role of defaults, that is, concepts that, modulo consistency, apply to each individual. Hence, starting with $\langle \mathcal{T}, \mathcal{B} \rangle$, we apply the following steps.

Step 1. Define $\mathcal{B}' = \mathcal{B} \cup \{C \sqcap \neg D \mathrel{|\!\sim} \bot \mid C \sqsubseteq D \in \mathcal{T}\}$. Now our agent is characterized by the pair $\langle \emptyset, \mathcal{B}' \rangle$.

Step 2. Define $\Delta_{\mathcal{B}'} = \{\top \sqsubseteq C \rightarrow D \mid C \mathrel{|\!\sim} D \in \mathcal{B}'\}$, and define a set $\mathfrak{A}_{\mathcal{B}'}$ as the set of the antecedents of the conditionals in $\mathcal{B}'$, *i.e.* $\mathfrak{A}_{\mathcal{B}'} = \{C \mid C \mathrel{|\!\sim} D \in \mathcal{B}'\}$.

Step 3. We determine the exceptionality ranking of the sequents in $\mathcal{B}'$ using the set of the antecedents $\mathfrak{A}_{\mathcal{B}'}$ and the materializations in $\Delta_{\mathcal{B}'}$, where a concept C is *exceptional* w.r.t. a set of sequents $\mathcal{D}$ iff $\Delta_{\mathcal{D}} \models \top \sqsubseteq \neg C$. The steps are the same of the propositional case (**Steps 3.1 – 3.2**), we just replace the expression $\Delta_{\mathcal{D}} \models \neg C$ with the expression $\Delta_{\mathcal{D}} \models \top \sqsubseteq \neg C$. In this way we define a ranking function r.

Step 4. From $\Delta_{\mathcal{B}'}$ and the ranking function r we obtain (i) that (**Step 4.1.**) we can verify if the conceptual system of the agent is consistent by checking the consistency of $\Delta_{\mathcal{B}'}$, and (ii) (**Steps 4.2.-4.3.**) we can define the real background theory and the defeasible information of the agent, respectively the sets $\widetilde{\mathcal{T}}$ and $\widetilde{\mathcal{B}}$ as:

$$\widetilde{\mathcal{T}} = \{\top \sqsubseteq \neg C \mid C \sqsubset D \in \mathcal{B}' \text{ and } r(C \sqsubset D) = \infty\}$$
$$\widetilde{\mathcal{B}} = \{C \sqsubset D \mid C \sqsubset D \in \mathcal{B}' \text{ and } r(C \mathrel{|\!\sim} D) < \infty\} .$$

From $\widetilde{\mathcal{T}}$ and $\widetilde{\mathcal{B}}$ we define the correspondent sets of concepts $\Phi = \{\neg C \mid \top \sqsubseteq \neg C \in \widetilde{\mathcal{T}}\}$ and $\Delta = \{C \to D \mid C \sqsubset D \in \widetilde{\mathcal{B}}\}$.

Step 5. Now, obtained $\langle \Phi, \Delta \rangle$ and the ranking value of the elements of $\widetilde{\mathcal{B}}$ and, consequently, of Δ (assume $r(\widetilde{\mathcal{B}}) = k$), we can determine the seriousness ordering on the subsets of Δ. The procedure is the same as for the propositional case, that is, (i) we associate to every subset $\mathcal{D}$ of Δ a string $\langle n_0, \ldots, n_k \rangle$ with $n_i = \mid \mathcal{D} \cap \Delta^{k-i} \mid$, and we obtain a lexicographic order '$>$' between the strings. Then we define the seriousness ordering $\prec$ between the subsets of Δ as

$$\mathcal{D} \prec \mathcal{E} \text{ iff } \langle n_0, \ldots, n_k \rangle_{\mathcal{D}} > \langle n_0, \ldots, n_k \rangle_{\mathcal{E}}$$

for every pair of subsets $\mathcal{D}$ and $\mathcal{E}$ of Δ.

Hence, we obtain an analogous of the procedure defined for the propositional case by substituting the conceptual system $\langle \mathcal{T}, \mathcal{B} \rangle$ with the pair $\langle \Phi, \Delta \rangle$.

Closure Operation over Concepts. Consider the pair $\langle \Phi, \Delta \rangle$. Now we specify the notion of lexicographic closure over the concepts, that is, a relation $\mathrel{|\!\sim}^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle}$ that tells us what presumably follows from a finite set of concepts Γ. Again, we define for a set of premises Γ the set of the most serious subsets of Δ that are consistent with Γ and Φ.

$$\mathfrak{D}_{\Gamma} = \min_{\prec}\{\mathcal{D} \subseteq \Delta \mid \not\models \bigsqcap \Gamma \sqcap \bigsqcap \Phi \sqcap \bigsqcap \mathcal{D} \sqsubseteq \bot\}$$

Having obtained $\mathfrak{D}_{\Gamma}$, the lexicographic closure is defined as follows:

$$\Gamma \mathrel{|\!\sim}^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle} E \textit{ iff } \models \bigsqcap \Gamma \sqcap \bigsqcap \Phi \sqcap \bigsqcap \mathcal{D} \sqsubseteq E \textit{ for every } \mathcal{D} \in \mathfrak{D}_{\Gamma} \ .$$

We can prove two main properties characterizing the proposition lexicographic closure and respected by $\mathrel{|\!\sim}^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle}$: (i) $\mathrel{|\!\sim}^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle}$ is a rational consequence relation, and (ii) $\mathrel{|\!\sim}^{l}_{\langle \mathcal{T}, \mathcal{B} \rangle}$ extends the rational closure.

Proposition 1. $\mathrel{|\!\sim}_{\langle \widetilde{\mathcal{T}}, \widetilde{\mathcal{B}} \rangle}$ *is a rational consequence relation validating* $\mathcal{K} = \langle \mathcal{T}, \mathcal{B} \rangle$.

This can be shown by noting that the analogous properties of the propositional rational consequence relation are satisfied, namely:

$$\text{(REF)}\ \ C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} C$$

$$\text{(LLE)}\ \frac{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E \qquad \models C = D}{D \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E} \qquad \text{(RW)}\ \frac{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} D \qquad \models D \sqsubseteq E}{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E}$$

$$\text{(CT)}\ \frac{C \sqcap D \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E \qquad C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} D}{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E} \qquad \text{(CM)}\ \frac{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E \qquad C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} D}{C \sqcap D \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E}$$

$$\text{(OR)}\ \frac{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E \qquad D \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E}{C \sqcup D \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} E} \qquad \text{(RM)}\ \frac{C \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} D \qquad C \not\mid\!\sim_{\langle \mathcal{T},\Delta\rangle} \neg E}{C \sqcap E \mid\!\sim_{\langle \mathcal{T},\Delta\rangle} D}$$

For the rational closure in $\mathcal{ALC}$, we refer to the construction presented in [9], Section 3. We indicate by $\mid\!\sim^{r}_{\langle \mathcal{T},\mathcal{B}\rangle}$ the consequence relation defined there.

Proposition 2. *The lexicographic closure extends the rational closure,* i.e. $\mid\!\sim^{r}_{\langle \mathcal{T},\mathcal{B}\rangle} \subseteq \mid\!\sim^{l}_{\langle \mathcal{T},\mathcal{B}\rangle}$ *for every pair* $\langle \mathcal{T}, \mathcal{B}\rangle$.

To prove this it is sufficient to check that, given a set of premises Γ and a pair $\langle \Phi, \Delta\rangle$, each of the sets in $\mathfrak{D}_\Gamma$ classically implies the default information that would be associated to Γ in its rational closure, as defined in [9].
Let us work out the analogue of Example 3 in the DL context.

Example 5. Consider the KB of Example 4 without the ABox. Hence, we start with $\mathcal{K} = \langle \mathcal{T}, \mathcal{B}\rangle$. Then $\mathcal{K}$ is changed into $\mathcal{B}' = \{P \sqcap \neg B \sqsubseteq \bot, I \sqcap Fi \sqsubseteq \bot, P \sqsubseteq \neg F, B \sqsubseteq F, B \sqsubseteq T, P \sqsubseteq \forall Prey.Fi, B \sqsubseteq \forall Prey.I\}$. The set of the materializations of $\mathcal{B}'$ is $\Delta_{\mathcal{B}'} = \{\top \sqsubseteq P \wedge \neg B \rightarrow \bot, \top \sqsubseteq I \sqcap Fi \rightarrow \bot, \top \sqsubseteq P \rightarrow \neg F, \top \sqsubseteq B \rightarrow F, \top \sqsubseteq B \rightarrow T, \top \sqsubseteq P \rightarrow \forall Prey.Fi, \top \sqsubseteq B \rightarrow \forall Prey.I\}$, with $\mathfrak{A}_{\mathcal{B}'} = \{P \wedge \neg B, I \sqcap Fi, P, B\}$. Following the procedure at **Step 3**, we obtain the ranking values of every inclusion axiom in $\mathcal{B}'$: namely, $r(B \sqsubseteq F) = r(B \sqsubseteq T) = r(B \sqsubseteq \forall Prey.I) = 0$; $r(P \sqsubseteq \neg F) = r(P \sqsubseteq \forall Prey.Fi) = 1$ and $r(P \sqcap \neg B \sqsubseteq \bot) = r(I \sqcap Fi \sqsubseteq \bot) = \infty$. From such a ranking, we obtain a background theory $\Phi = \{\neg(P \wedge \neg B), \neg(I \sqcap Fi)\}$, and a default set $\Delta = \Delta^0 \cup \Delta^1$, with

$$\Delta^0 = \{B \rightarrow F, B \rightarrow T, B \rightarrow \forall Prey.I\}$$
$$\Delta^1 = \{P \rightarrow \neg F, P \rightarrow \forall Prey.Fi\}\ .$$

To check if $P \mid\!\sim^{l}_{\langle \mathcal{T},\mathcal{B}\rangle} T$, we have to find the most serious subsets of Δ that are consistent with P and the concepts in Φ (*i.e.* the most serious subsets $\mathcal{D}$ of Δ s.t. $\not\models \bigsqcap \Gamma \sqcap \bigsqcap \Phi \sqcap \bigsqcap \mathcal{D} \sqsubseteq \bot$). Turns out that there is only one, $\mathcal{D} = \{P \rightarrow \neg F, P \rightarrow \forall Prey.Fi, B \rightarrow T\}$, and $\models P \sqcap \bigsqcap \Phi \sqcap \bigsqcap \mathcal{D} \sqsubseteq T$.
It is easy to check that we obtain the analogue sequents as in the propositional case and avoid the same undesirable ones. Moreover we can derive also sequents connected to the roles, such as $B \mid\!\sim^{l}_{\langle \mathcal{T},\mathcal{B}\rangle} \forall Prey.\neg Fi$ and $P \mid\!\sim^{l}_{\langle \mathcal{T},\mathcal{B}\rangle} \forall Prey.\neg I$. □

We do not have yet a proper proof, but we conjecture that the decision procedure should be EXPTIME-complete also in $\mathcal{ALC}$.

Closure Operation over Individuals. Now we will pay attention on how to apply the lexicographic closure to the ABox. Unfortunately, the application of the lexicographic

closure to the ABox results into a really more complicated procedure than in the case of rational closure, as presented in the last paragraph of [9], Section 3. Assume that we have already transformed our conceptual system $\langle \mathcal{T}, \mathcal{B} \rangle$ into a pair $\langle \widetilde{\mathcal{T}}, \widetilde{\mathcal{B}} \rangle$, and eventually into a pair $\langle \Phi, \Delta \rangle$. In particular, dealing with the ABox, we assume that we start with a knowledge base $\mathcal{K} = \langle \mathcal{A}, \widetilde{\mathcal{T}}, \Delta \rangle$. We would like to infer whether a certain individual a is presumably an instance of a concept C or not. The basic idea remains to associate to every individual in $\mathcal{A}$ every default information from Δ that is consistent with our knowledge base, respecting the seriousness ordering of the subsets of Δ. As we will see, the major problem to be addressed here is that we cannot obtain anymore a unique lexicographic extension of the KB.

Example 6. Consider $\mathcal{K} = \langle \mathcal{A}, \emptyset, \Delta \rangle$, with $\mathcal{A} = \{(a, b):R\}$ and $\Delta = \{A \sqcap \forall R.\neg A\}$. Informally, if we apply the default to a first, we get $b{:}\neg A$ and we cannot apply the default to b, while if we apply the default to b first, we get $b{:}A$ and we cannot apply the default to a. Hence, we may have *two* extensions. □

The possibility of multiple extensions is due to the presence of the roles, that allow the transmission of information from an individual to another; if every individual was 'isolated', without role-connections, then the addition of the default information to each individual would have been a 'local' problem, treatable without considering the concepts associated to the other individuals in the domain, and the extension of the knowledge base would have been unique. On the other hand, while considering a specific individual, the presence of the roles forces to consider also the information associated to other individuals in order to maintain the consistency of the knowledge base, and, as shown in example 6, the addition of default information to one individual could prevent the association of default information to another.

We assume that $\langle \mathcal{A}, \mathcal{T} \rangle$ is consistent, *i.e.* $\langle \mathcal{A}, \mathcal{T} \rangle \not\models a{:}\bot$, for any a. With $\mathcal{O}_\mathcal{A}$ we indicate the individuals occurring in $\mathcal{A}$. Given $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \Delta \rangle$, we say that a knowledge base $\widetilde{\mathcal{K}} = \langle \mathcal{A}_\Delta, \mathcal{T} \rangle$ is a *default extension* of $\mathcal{K}$ iff

- $\widetilde{\mathcal{K}}$ is classically consistent and $\mathcal{A} \subseteq \mathcal{A}_\Delta$.
- For any $a \in \mathcal{O}_\mathcal{A}$, $a{:}C \in \mathcal{A}_\Delta \setminus \mathcal{A}$ iff $C = \bigsqcap \mathcal{D}$ for some $\mathcal{D} \subset \Delta$ s.t. $\langle \mathcal{A}_\Delta \cup \{a{:}\mathcal{D}'\}, \mathcal{T} \rangle \models \bot$ for every $\mathcal{D}' \subseteq \Delta$, with $\mathcal{D}' \prec \mathcal{D}$.

Essentially, we assign to each individual $a \in \mathcal{O}_\mathcal{A}$ one of the most serious default sets that are consistent with the ABox.

Example 7. Referring to Example 6, consider $\mathcal{K} = \langle \mathcal{A}, \emptyset, \Delta \rangle$, with $\mathcal{A} = \{(a, b) : R\}$ and $\Delta = \{A \sqcap \forall R.\neg A, \top\}$. Then we have two default-assumption extensions, namely $\widetilde{\mathcal{K}}_1 = \langle \mathcal{A} \cup \{a{:}A, a{:}\forall R.\neg A\}, \emptyset \rangle$ and $\widetilde{\mathcal{K}}_2 = \langle \mathcal{A} \cup \{b{:}A, b{:}\forall R.\neg A\}, \emptyset \rangle$. □

A procedure to obtain a set A_s of default extensions is as follows:

(i) fix a linear order $s = \langle a_1, \ldots, a_m \rangle$ of the individuals in $\mathcal{O}_\mathcal{A}$, and let $\mathsf{A}_s^0 = \{\mathcal{A}\}$.

Now, for every a_i, $1 \leq i \leq m$, do:

(ii) for every element $\mathcal{X}$ of A_s^{i-1}, find the set all the $\prec$-minimal default sets $\{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, s.t. $\mathcal{D}_j \subseteq \Delta$ and $\mathcal{X} \cup \{a_i{:}\bigsqcap \mathcal{D}_j\}$ is consistent $(1 \leq j \leq n)$;

(iii) Define a new set A_s^i containing all the sets $\mathcal{X} \cup \{a_i{:}\bigsqcap \mathcal{D}_j\}$ obtained at the point (ii).

(iv) Move to the next individual a_{i+1}.

(v) Once the points (ii)-(iv) have been applied to all the individuals in the sequence $s = \langle a_1, \ldots, a_m \rangle$, set $\mathsf{A}_s = \mathsf{A}_s^m$, where A_s^m is the final set obtained at the end of the procedure.

It can be shown that

Proposition 3. *An Abox $\mathcal{A}'$ is a default extension of $\mathcal{K} = \{\mathcal{A}, \Delta\}$ iff it is in the set A_s obtained by some linear ordering s on $\mathcal{O}_\mathcal{A}$ and the above procedure.*

For instance, related to Example 7, $\widetilde{\mathcal{K}}_1$ is obtained from the order $\langle a, b \rangle$, while $\widetilde{\mathcal{K}}_2$ is obtained from the order $\langle b, a \rangle$.

Example 8. Refer to Example 5, and let $\mathcal{K} = \{\mathcal{A}, \mathcal{T}, \Delta\}$, where $\mathcal{A} = \{a{:}P,\ b{:}B,\ (a,c){:}Prey,\ (b,c){:}Prey\}$, $\mathcal{T} = \{P \sqsubseteq B, I \sqsubseteq \neg Fi\}$, $\Delta = \{B \rightarrow F, B \rightarrow T, B \rightarrow \forall Prey.I, P \rightarrow \neg F, P \rightarrow \forall Prey.Fi\}$. If we consider an order where a is before b then we associate $\mathcal{D} = \{B \rightarrow T, P \rightarrow \neg F, P \rightarrow \forall Prey.Fi\}$ to a, and consequently c is presumed to be a fish and we are prevented in the association of $B \rightarrow \forall Prey.I$ to b. If we consider b before a, c is not a fish and we cannot apply $P \rightarrow \forall Prey.Fi$ to a. □

If we fix a priori a linear order s on the individuals, we may define a consequence relation depending on the default extensions generated from it, *i.e.* the sets of defaults in A_s: we say that $a{:}C$ is a *defeasible consequence* of $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \Delta \rangle$ w.r.t. s, written $\mathcal{K} \mid\!\sim^l_s a{:}C$, iff $\mathcal{A}' \models a{:}C$ for every $\mathcal{A} \in \mathsf{A}_s$.

For instance, related to Example 7 and order $s_1 = \langle a, b \rangle$, we may infer that $\mathcal{K} \mid\!\sim^l_{s_1} a{:}A$, while with order $s_2 = \langle b, a \rangle$, we may infer that $\mathcal{K} \mid\!\sim^l_{s_2} b{:}A$.

The interesting point of such a consequence relation is that it satisfies the properties of a *rational* consequence relation in the following way.

$$REF_{DL} \quad \langle \mathcal{A}, \Delta \rangle \mid\!\sim_s a{:}C \text{ for every } a{:}C \in \mathcal{A}$$

$$LLE_{DL} \quad \frac{\langle \mathcal{A} \cup \{b{:}D\}, \Delta \rangle \mid\!\sim_s a{:}C \quad \models D = E}{\langle \mathcal{A} \cup \{b{:}E\}, \Delta \rangle \mid\!\sim_s a{:}C}$$

$$RW_{DL} \quad \frac{\langle \mathcal{A}, \Delta \rangle \mid\!\sim_s a{:}C \quad \models C \sqsubseteq D}{\langle \mathcal{A}, \Delta \rangle \mid\!\sim_s a{:}D}$$

$$CT_{DL} \quad \frac{\langle \mathcal{A} \cup \{b{:}D\}, \Delta \rangle \mid\!\sim_s a{:}C \quad \langle \mathcal{A}, \Delta \rangle \mid\!\sim_s b{:}D}{\langle \mathcal{A}, \Delta \rangle \mid\!\sim_s a{:}C}$$

$$CM_{DL} \quad \frac{\langle \mathcal{A}, \Delta \rangle \mid\!\sim_s a{:}C \quad \langle \mathcal{A}, \Delta \rangle \mid\!\sim_s b{:}D}{\langle \mathcal{A} \cup \{b{:}D\}, \Delta \rangle \mid\!\sim_s a{:}C}$$

$$OR_{DL} \quad \frac{\langle \mathcal{A} \cup \{b{:}D\}, \Delta \rangle \mid\!\sim_s a{:}C \quad \langle \mathcal{A} \cup \{b{:}E\}, \Delta \rangle \mid\!\sim_s a{:}C}{\langle \mathcal{A} \cup \{b{:}D \sqcup E\}, \Delta \rangle \mid\!\sim_s a{:}C}$$

$$RM_{DL} \quad \frac{\langle \mathcal{A}, \Delta \rangle \mid\!\sim_s a{:}C \quad \langle \mathcal{A}, \Delta \rangle \not\mid\!\sim_s b{:}\neg D}{\langle \mathcal{A} \cup \{b{:}D\}, \Delta \rangle \mid\!\sim_s a{:}C}$$

We can show that

Proposition 4. *Given $\mathcal{K}$ and a linear order s of the individuals in $\mathcal{K}$, the consequence relation $\mid\!\sim^l_s$ satisfies the properties* $\mathrm{REF_{DL}} - \mathrm{RM_{DL}}$.

4 Conclusions

In this paper we have proposed an extension of a main non-monotonic construction, the lexicographic closure (see [18]), for the DL $\mathcal{ALC}$. This work carries forward the approach presented in [9], where the adaptation of the rational closure in $\mathcal{ALC}$ is presented. Here we have first presented the procedure at the propositional level, and then we have adapted it for $\mathcal{ALC}$, first considering only the conceptual level, the information contained in the TBox, and then considering also the particular information about the individuals, the ABox, assuming we are working with unfoldable KB.
It is straightforward to see that, while the procedure defined for the TBox is simple and well-behaved, the procedure for the ABox is really more complex than the one for the rational closure presented in [9].
Besides checking the exact costs of these procedures from the computational point of view and checking for which other DL formalisms we can apply them, we conjecture that a semantical characterization of the above procedures can be obtained using the kind of semantical constructions presented in [8].

References

1. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
2. Franz Baader and Bernhard Hollunder. How to prefer more specific defaults in terminological default logic. In *In Proceedings of the IJCAI*, pages 669–674. Morgan Kaufmann Publishers, 1993.
3. Alexander Bochman. *A logical theory of nonmonotonic inference and belief change*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.
4. P. Bonatti, C. Lutz, and F. Wolter. The complexity of circumscription in description logic. *Journal of Artificial Intelligence Research*, 35:717–773, 2009.
5. Piero A. Bonatti, Marco Faella, and Luigi Sauro. Defeasible inclusions in low-complexity dls. *J. Artif. Intell. Res. (JAIR)*, 42:719–764, 2011.
6. Piero A. Bonatti, Marco Faella, and Luigi Sauro. On the complexity of el with defeasible inclusions. In *IJCAI-11*, pages 762–767. IJCAI/AAAI, 2011.
7. Gerhard Brewka and D Sankt Augustin. The logic of inheritance in frame systems. In *IJCAI-87*, pages 483–488. Morgan Kaufmann Publishers, 1987.
8. K. Britz, T. Meyer, and I. Varzinczak. Semantic foundation for preferential description logics. In D. Wang and M. Reynolds, editors, *Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence*, number 7106 in LNAI, pages 491–500. Springer, 2011.
9. Giovanni Casini and Umberto Straccia. Rational closure for defeasible description logics. In Tomi Janhunen and Ilkka Niemelä, editors, *JELIA*, volume 6341 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2010.
10. Giovanni Casini and Umberto Straccia. Defeasible inheritance-based description logics. In *IJCAI-11*, pages 813–818, 2011.
11. Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Description logics of minimal knowledge and negation as failure. *ACM Trans. Comput. Log.*, 3(2):177–225, 2002.
12. Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors. *Handbook of logic in artificial intelligence and logic programming (vol. 3): nonmonotonic reasoning and uncertain reasoning*. Oxford University Press, Inc., New York, NY, USA, 1994.

13. Laura Giordano, Valentina Gliozzi, Nicola Olivetti, and Gian Luca Pozzato. Reasoning about typicality in low complexity dls: The logics el$^{\bot}$t$_{\min}$ and dl-lite$_{\mathrm{c}}$ t$_{\min}$. In *IJCAI*, pages 894–899. IJCAI/AAAI, 2011.
14. Laura Giordano, Nicola Olivetti, Valentina Gliozzi, and Gian Luca Pozzato. Alc + t: a preferential extension of description logics. *Fundam. Inform.*, 96(3):341–372, 2009.
15. S. Grimm and P. Hitzler. A preferential tableaux calculus for circumscriptive $\mathcal{ALCO}$. In *RR-09*, number 5837 in LNCS, pages 40–54, Berlin, Heidelberg, 2009. Springer-Verlag.
16. Sarit Kraus, Daniel Lehmann, and Menachem Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intell.*, 44(1-2):167–207, 1990.
17. Daniel Lehmann and Menachem Magidor. What does a conditional knowledge base entail? *Artif. Intell.*, 55(1):1–60, 1992.
18. Daniel J. Lehmann. Another perspective on default reasoning. *Ann. Math. Artif. Intell.*, 15(1):61–82, 1995.
19. David Makinson. General patterns in nonmonotonic reasoning. In *Handbook of logic in artificial intelligence and logic programming: nonmonotonic reasoning and uncertain reasoning*, volume 3, pages 35–110. Oxford University Press, Inc., New York, NY, USA, 1994.
20. David Poole. A logical framework for default reasoning. *Artif. Intell.*, 36(1):27–47, 1988.
21. Joachim Quantz and Veronique Royer. A preference semantics for defaults in terminological logics. In *KR-92*, pages 294–305. Morgan Kaufmann, Los Altos, 1992.
22. U. Straccia. Default inheritance reasoning in hybrid kl-one-style logics. *IJCAI-93*, pages 676–681, 1993.

A normal form for hypergraph-based module extraction for $\mathcal{SROIQ}$

Riku Nortje, Katarina Britz, and Thomas Meyer

Center for Artificial Intelligence Research, University of KwaZulu-Natal and CSIR Meraka Institute, South Africa
Email: nortjeriku@gmail.com; {arina.britz;tommie.meyer}@meraka.org.za

Abstract. Modularization is an important part of the modular design and maintenance of large scale ontologies. Syntactic locality modules, with their desirable model theoretic properties, play an ever increasing role in the design of algorithms for modularization, partitioning and reasoning tasks such as classification. It has been shown that, for the DL $\mathcal{EL}^+$, the syntactic locality module extraction problem is equivalent to the reachability problem for hypergraphs. In this paper we investigate and introduce a normal form for the DL $\mathcal{SROIQ}$ which allows us to map any $\mathcal{SROIQ}$ ontology to an equivalent hypergraph. We then show that standard hyperpath search algorithms can be used to extract modules similar to syntactic locality modules for $\mathcal{SROIQ}$ ontologies.

1 Introduction

The advent of the semantic web presupposes a significant increase in the size of ontologies, their distributive nature and the requirement for fast reasoning algorithms. Modularization techniques not only play an increasingly important role in the design and maintenance of large-scale distributed ontologies, but also in the design of algorithms that increase the efficiency of reasoning tasks such as subsumption testing and classification [11, 1].

Extracting minimal modules is computationally expensive and even undecidable for expressive DLs [2, 3]. Therefore, the use of approximation techniques and heuristics play an important role in the effective design of algorithms. Syntactic locality [2, 3], because of its excellent model theoretic properties, has become an ideal heuristic and is widely used in a diverse set of algorithms [11, 1, 4].

Suntisrivaraporn [11] showed that, for the DL $\mathcal{EL}^+$, $\bot$-locality module extraction is equivalent to the reachability problem in directed hypergraphs. Nortjé et al. [9, 10] extended the reachability problem to include $\top$-locality and introduced bidirectional reachability modules as a subset of $\bot\top$ modules.

In this paper we introduce a normal form for the DL $\mathcal{SROIQ}$, which allows us to map any $\mathcal{SROIQ}$ ontology to an equivalent syntactic locality preserving hypergraph. We show that, given this mapping, the extraction of $\bot$-locality modules is equivalent to the extraction of all B-hyperpaths, $\top$-locality is similar to extracting all F-hyperpaths and $\bot\top^*$ modules to that of extracting frontier graphs. These similarities demonstrate a unique relationship between reasoning

tasks, based on syntactic locality, for $\mathcal{SROIQ}$ ontologies, and standard well studied hypergraph algorithms.

2 Preliminaries

2.1 Hypergraphs

Hypergraphs are a generalization of graphs and have been extensively studied since the 1970s as a powerful tool for modelling many problems in Discrete Mathematics. In this paper we adapt the definitions of hypergraphs and hyperpaths from [8, 12].

A (directed) hypergraph is a pair $\mathcal{H} = \langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is a finite set of nodes, $\mathcal{E} \subseteq 2^{\mathcal{V}} \times 2^{\mathcal{V}}$ is the set of hyperedges such that for every $e = (T(e), H(e)) \in \mathcal{E}$, $T(e) \neq \emptyset$, $H(e) \neq \emptyset$, and $T(e) \cap H(e) = \emptyset$. A hypergraph $\mathcal{H}' = \langle \mathcal{V}', \mathcal{E}' \rangle$ is a subhypergraph of $\mathcal{H}$ if $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. A hyperedge e is a *B-hyperedge* if $|H(e)| = 1$. A *B-hypergraph* is a hypergraph such that each hyperedge is a B-hyperedge. A hyperedge e is an *F-hyperedge* if $|T(e)| = 1$. An *F-hypergraph* is a hypergraph such that each hyperedge is an F-hyperedge. A *BF-hypergraph* is a hypergraph for which every edge is either a B- or an F-hyperedge.

Let $e = (T(e), H(e))$ be a hyperedge in some directed hypergraph $\mathcal{H}$. Then, $T(e)$ is known as the *tail* of e and $H(e)$ is known as the *head* of e. Given a directed hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, its symmetric image $\overline{\mathcal{H}}$ is a directed hypergraph defined as: $\mathcal{V}(\overline{\mathcal{H}}) = \mathcal{V}(\mathcal{H})$ and $\mathcal{E}(\overline{\mathcal{H}}) = \{(H, T) \mid (T, H) \in \mathcal{E}(\mathcal{H})\}$. We denote by $BS(v) = \{e \in \mathcal{E} \mid v \in H(e)\}$ and $FS(v) = \{e \in \mathcal{E} \mid v \in T(e)\}$ respectively the *backward star* and *forward star* of a node v. Let n and m be the number of nodes and hyperedges in a hypergraph $\mathcal{H}$. We define the size of $\mathcal{H}$ as $size(\mathcal{H}) = |\mathcal{V}| + \sum_{e \in \mathcal{E}} (|T(e)| + |H(e)|)$.

A simple path $\prod_{st}$ from $s \in \mathcal{V}(\mathcal{H})$ to $t \in \mathcal{V}(\mathcal{H})$ in $\mathcal{H}$ is a sequence $(v_1, e_1, v_2, e_2, ..., v_k, e_k, v_{k+1})$ consisting of distinct nodes and hyperedges such that $s = v_1$, $t = v_{k+1}$ and for every $1 \leq i \leq k$, $v_i \in T(e_i)$ and $v_{i+1} \in H(e_i)$. If in addition $t \in T(e_1)$ then $\prod_{st}$ is a simple cycle. A simple path is *cycle free* if it does not contain any subpath that is a simple cycle.

A node s is B-connected to itself. If there is a hyperedge e such that all nodes $v_i \in T(e)$ are B-connected to s, then every $v_j \in H(e)$ is B-connected to s. A B-hyperpath from $s \in \mathcal{V}(\mathcal{H})$ to $t \in \mathcal{V}(\mathcal{H})$ is a minimal subhypergraph of $\mathcal{H}$ where t is B-connected to s. An F-hyperpath $\prod_{st}$ from $s \in \mathcal{V}(\mathcal{H})$ to $t \in \mathcal{V}(\mathcal{H})$ in $\mathcal{H}$ is a subhyperpath of $\mathcal{H}$ such that $\overline{\prod}_{st}$ is a B-hyperpath from t to s in $\overline{\mathcal{H}}$. A BF-hyperpath from $s \in \mathcal{V}(\mathcal{H})$ to $t \in \mathcal{V}(\mathcal{H})$ in $\mathcal{H}$ is a minimal (in the inclusion sense) subhyperpath of $\mathcal{H}$ such that it is simultaneously both a B-hyperpath and an F-hyperpath from s to t in $\mathcal{H}$. We note that every hypergraph $\mathcal{H}$ can be transformed to a BF-hypergraph $\mathcal{H}'$ by replacing each hyperedge $e = (T(e), H(e))$ with the two hyperedges $e_1 = (T(e), \{n_v\}), e_2 = (\{n_v\}, H(e))$ where n_v is a new node.

Algorithm 1 (Visiting a hypergraph [8])

`Procedure Bvisit(s,`$\mathcal{H}$`)`	`Procedure Fvisit(t,`$\mathcal{H}$`)`
1 : **for each** $u \in \mathcal{V}$ **do** $blabel(u) := false$;	**for each** $u \in \mathcal{V}$ **do** $flabel(u) := false$;
2 : **for each** $e \in \mathcal{E}$ **do** $T(e) := 0$;	**for each** $e \in \mathcal{E}$ **do** $T(e) := 0$;
3 : $Q := \{s\}; blabel(s) := true$;	$Q := \{t\}; flabel(t) := true$;
4 : **while** $Q \neq \emptyset$ **do**	**while** $Q \neq \emptyset$ **do**
5 : **select and remove** $u \in Q$;	**select and remove** $u \in Q$;
6 : **for each** $e \in FS(u)$ **do**	**for each** $e \in BS(u)$ **do**
7 : $T(e) := T(e) + 1$;	$H(e) := H(e) + 1$;
8 : **if** $T(e) := \|Tail(e)\|$ ***then***	**if** $H(e) := \|Head(e)\|$ ***then***
9 : **for each** $v \in Head(e)$ **do**	**for each** $v \in Tail(e)$ **do**
10 : **if** $blabel(v) = false$ **then**	**if** $flabel(v) = false$ **then**
11 : $blabel(v) = true$	$flabel(v) = true$
12 : $Q := Q \cup \{v\}$	$Q := Q \cup \{v\}$

Given some node s, Algorithm 1 can be used to find all B-connected or F-connected nodes to s in $O(size(\mathcal{H}))$ time. Here, the set of all B-hyperpaths from s and F-hyperpaths to t are respectively represented by all those nodes n such that $blabel(n) = true$ or $flabel(n) = true$, as well as the edges connecting those nodes.

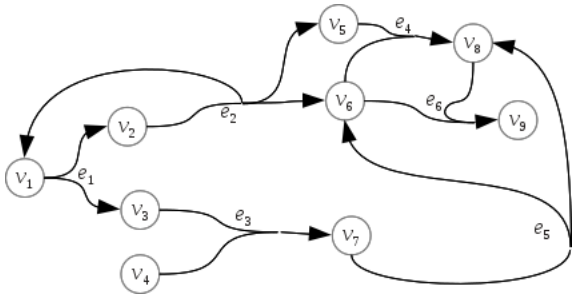

Fig. 1. Example hypergraph $\mathcal{H}_1$

Example 1. In Figure 1 we have $\mathcal{H}_1 = (\mathcal{V}_1, \mathcal{E}_1)$, with $\mathcal{V}_1 = \{v_1, ..., v_9\}$ and $\mathcal{E}_1 = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ such that $e_1 = (\{v_1\}, \{v_2, v_3\})$, $e_2 = (\{v_2\}, \{v_1, v_5, v_6\})$, $e_3 = (\{v_3, v_4\}, \{v_7\})$, $e_4 = (\{v_5, v_6\}, \{v_8\})$, $e_5 = (\{v_7\}, \{v_6, v_8\})$ and $e_6 = (\{v_6, v_8\}, \{v_9\})$. The directed hypergraph $\mathcal{G}_1$ with nodes $\mathcal{V}(\mathcal{G}_1) = \{v_1, v_2, v_3, v_5, v_6, v_8, v_9\}$ and $\mathcal{E}(\mathcal{G}_1) = \{e_1, e_2, e_4, e_6\}$ is a B-hyperpath from v_1 to v_9 in $\mathcal{H}_1$. The hypergraph $\mathcal{G}_2$ with $\mathcal{V}(\mathcal{G}_2) = \{v_3, v_4, v_6, v_7, v_8, v_9\}$ and $\mathcal{E}(\mathcal{G}_2) = \{e_3, e_5, e_6\}$ is an F-hyperpath from v_3 to v_9 in $\mathcal{H}_1$. The hypergraph $\mathcal{G}_3$ with $\mathcal{V}(\mathcal{G}_3) = \{v_6, \dots v_9\}$ and $\mathcal{E}(\mathcal{G}_3) = \{e_5, e_6\}$ is a BF-hyperpath from v_7 to v_9 in $\mathcal{H}_1$.

Definition 1. *Given a hypergraph $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, the* frontier *graph $\mathcal{H}' = (\mathcal{V}', \mathcal{E}', s, t)$ of $\mathcal{H}$, such that $\mathcal{V}' \subseteq \mathcal{V}$, $\mathcal{E}' \subseteq \mathcal{E}$, $s, t \in \mathcal{V}$, is the maximal (in the inclusion sense) BF-graph in which (1) s and t are the origin and destination nodes, (2) if $v \in \mathcal{V}'$ then v is B-connected to s, and t is F-connected to v in $\mathcal{H}'$.*

Algorithm 2 (Frontier graph Extraction Algorithm [8])

`Procedure frontier(`$\mathcal{H}, \mathcal{H}', s, t$`)`

$1: \mathcal{H}' := \mathcal{H}$; change := true
$2:$ **while** $change = true$ **do**
$3:$ $change = false$
$3:$ $\mathcal{H}' = Bvisit(s, \mathcal{H}')$; $\mathcal{H}' = Fvisit(t, \mathcal{H}')$
$4:$ **for each** $v \in \mathcal{V}'$
$5:$ **if** $blabel(v) = false$ **or** $flabel(v) = false$ **then**
$6:$ $change := true$
$7:$ $\mathcal{V}' = \mathcal{V}' - \{v\}$; $\mathcal{E}' = \mathcal{E}' - FS(v) - BS(v)$
$8:$ **if** $s \notin \mathcal{V}'$ **or** $t \notin \mathcal{V}'$ **then**
$9:$ $\mathcal{H}' := \emptyset; change := false;$

Algorithm 2 can be used to extract a frontier graph for any source and destination nodes and runs in $O(n\, size(\mathcal{H}))$ time.

2.2 The DL $\mathcal{SROIQ}$

In this section we give a brief introduction to the DL $\mathcal{SROIQ}$ [5, 7] with its syntax and semantics listed in Table 1. N_C, N_R and N_I denote disjoint sets of atomic concept names, atomic roles names and individual names. The set N_R includes the universal role. Well-formed formulas are created by combining concepts from the table by using the connectives $\neg, \sqcap, \sqcup$ etc.

Given $R_1 \circ \ldots \circ R_n \sqsubseteq R$, where $n \geqslant 1$ and $R_i, R \in N_R$, is a *role inclusion axiom* (RIA). A *role hierarchy* is a finite set of RIAs. Here $R_1 \circ \ldots \circ R_n$ denotes a composition of roles where R, R_i may also be an *inverse role* R^-. A role R is *simple* if it: (1) does not appear on the right-hand side of a RIA; (2) is the inverse of a simple role; or (3) appears on the right-hand side of a RIA only if the left-hand side consists entirely of simple roles. $Ref(R)$, $Irr(R)$ and $Dis(R, S)$, where R, S are roles other than U, are role assertions. A set of role assertions is simple w.r.t. a role-hierarchy H if each assertion $Irr(R)$ and $Dis(R, S)$ uses only simple roles w.r.t. H.

A strict partial order $\prec$ on N_R is a *regular order* if, and only if, for all roles R and S: $S \prec R$ iff $S^- \prec R$. Let $\prec$ be a regular order on roles. A RIA $w \sqsubseteq R$ is $\prec$-regular if, and only if, $R \in N_R$ and w has one of the following forms: (1) $R \circ R$, (2) R^-, (3) $S_1 \circ \ldots \circ S_n$, where each $S_i \prec R$, (4) $R \circ S_1 \circ \ldots \circ S_n$, where each $S_i \prec R$ and (5) $S_1 \circ \ldots \circ S_n \circ R$, where each $S_i \prec R$. A role hierarchy H is *regular* if there exists a regular order $\prec$ such that each RIA in H is $\prec$-regular.

An $RBox$ is a finite, regular role hierarchy H together with a finite set of role assertions simple w.r.t. H. If $a_1, \ldots, a_n$ are in N_I, then $\{a_1, \ldots, a_n\}$ is a nominal. N_o is the set of all nominals. The set of $\mathcal{SROIQ}$ *concept descriptions* is the smallest set such that: (1) $\bot, \top$, each $C \in N_C$, and each $o \in N_o$ is a concept description. (2) If C is a concept description, then $\neg C$ is a concept description. (3) If C and D are concept descriptions, R is a role description, S is a simple role description, and n is a non-negative integer, then the following are all concept descriptions: $(C \sqcap D), (C \sqcup D), \exists R.C, \forall R.C, \leqslant nS.C, \geqslant nS.C, \exists S.Self$.

Table 1. Syntax and semantics of $\mathcal{SROIQ}$

Concept	Syntax	Semantics
atomic concept	$C \in N_C$	$C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
individual	$A \in N_I$	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
nominal	$\{a_1, \ldots, a_n\}, a_i \in N_I$	$\{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$
role	$R \in N_R$	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	$R^-, R \in N_R$	$R^{-\mathcal{I}} = \{(y,x) \mid (x,y) \in R^{\mathcal{I}}\}$
universal role	U	$U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
role composition	$R_1 \circ \ldots \circ R_n$	$\{(x,z) \mid (x,y_1) \in R_1^{\mathcal{I}} \wedge (y_1,y_2) \in R_2^{\mathcal{I}} \wedge \ldots \wedge (y_n,z) \in R_{n+1}^{\mathcal{I}}\}$
top	$\top$	$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$
bottom	$\bot$	$\bot^{\mathcal{I}} = \emptyset$
negation	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C_1 \sqcap C_2$	$(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$
disjunction	$C_1 \sqcup C_2$	$(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$
exist restriction	$\exists R.C$	$\{x \mid (\exists y)[(x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}]\}$
value restriction	$\forall R.C$	$\{x \mid (\forall y)[(x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}]\}$
self restriction	$\exists R.Self$	$\{x \mid (x,x) \in R^{\mathcal{I}}\}$
atmost restriction	$\leqslant nR.C$	$\{x \mid \#\{y \mid (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leqslant n\}$
atleast restriction	$\geqslant nR.C$	$\{x \mid \#\{y \mid (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geqslant n\}$
Axiom	**Syntax**	**Semantics**
concept inclusion	$C_1 \sqsubseteq C_2$	$C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$
role inclusion	$R_1 \circ \ldots \circ R_n \sqsubseteq R$	$(R_1 \circ \ldots \circ R_n)^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
reflexivity	$Ref(R)$	$\{(x,x) \mid x \in \Delta^{\mathcal{I}}\} \subseteq R^{\mathcal{I}}$
irreflexivity	$Irr(R)$	$\{(x,x) \mid x \in \Delta^{\mathcal{I}}\} \cap R^{\mathcal{I}} = \emptyset$
disjointness	$Dis(R,S)$	$S^{\mathcal{I}} \cap R^{\mathcal{I}} = \emptyset$
class assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
inequality assertion	$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$
role assertion	$R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
negative role assertion	$\neg R(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \notin R^{\mathcal{I}}$

If C and D are concept description then $C \sqsubseteq D$ is a *general concept inclusion* (GCI) axiom. A *TBox* is a finite set of GCIs. If C is a concept description, $a, B \in N_I$, $R, S \in N_R$ with S a simple role description, then $C(a)$, $R(a,b)$, $\neg S(a,b)$, and $a \neq b$, are individual assertions. An $\mathcal{SROIQ}$ *ABox* is a finite set of individual assertions. All GCIs, RIAs, role assertions, and individual assertions are referred to as axioms. A $\mathcal{SROIQ}$-KB base is the union of a TBox, RBox and ABox.

2.3 Modules

Definition 2. (Module for the arbitrary DL $\mathcal{L}$) *Let $\mathcal{L}$ be an arbitrary description language, $\mathcal{O}$ an $\mathcal{L}$ ontology, and σ a statement formulated in $\mathcal{L}$. Then, $\mathcal{O}' \subseteq \mathcal{O}$ is a* module for σ in $\mathcal{O}$ *(a σ-*module in $\mathcal{O}$*) whenever: $\mathcal{O} \models \sigma$ if and only if $\mathcal{O}' \models \sigma$. We say that $\mathcal{O}'$ is a module for a signature $\boldsymbol{S}$ in $\mathcal{O}$ (an* **S**-module in $\mathcal{O}$*) if, for every $\mathcal{L}$ statement σ with Sig(σ) $\subseteq \boldsymbol{S}$, $\mathcal{O}'$ is a σ-module in $\mathcal{O}$.*

Definition 2 is sufficiently general so that any subset of an ontology preserving a statement of interest is considered a module, the entire ontology is therefore a module in itself. An important property of modules in terms of the modular reuse of ontologies is *safety* [2, 3]. Intuitively, a module conforms to a safety condition whenever an ontology $\mathcal{T}$ reuses concepts from an ontology $\mathcal{T}'$ in such a way so that it does not change the meaning of any of the concepts in $\mathcal{T}'$. This may be formalized in terms of the notion of conservative extensions:

Definition 3. (Conservative extension [3]) *Let $\mathcal{T}$ and $\mathcal{T}_1$ be two ontologies such that $\mathcal{T}_1 \subseteq \mathcal{T}$, and let $\mathcal{S}$ be a signature. Then (1) $\mathcal{T}$ is an $\mathcal{S}$-conservative extension of $\mathcal{T}_1$ if, for every α with $Sig(\alpha) \subseteq \mathcal{S}$, we have $\mathcal{T} \models \alpha$ iff $\mathcal{T}_1 \models \alpha$. (2) $\mathcal{T}$ is a conservative extension of $\mathcal{T}_1$ if $\mathcal{T}$ is an $\mathcal{S}$-conservative extension of $\mathcal{T}_1$ for $\mathcal{S} = Sig(\mathcal{T}_1)$.*

Definition 4. (Safety [3, 6]) *An ontology $\mathcal{T}$ is* safe *for $\mathcal{T}'$ if $\mathcal{T} \cup \mathcal{T}'$ is a conservative extension of $\mathcal{T}'$. Further let S be a signature. We say that $\mathcal{T}$ is safe for S if, for every ontology $\mathcal{T}'$ with $Sig(\mathcal{T}) \cap Sig(\mathcal{T}') \subseteq S$, we have that $\mathcal{T} \cup \mathcal{T}'$ is a conservative extension of $\mathcal{T}'$.*

Intuitively, given a set of terms, or seed signature, S, a S-module $\mathcal{M}$ based on deductive-*conservative extensions* is a minimal subset of an ontology $\mathcal{O}$ such that for all axioms α with terms only from S, we have that $\mathcal{M} \models \alpha$ if, and only if, $\mathcal{O} \models \alpha$, i.e., $\mathcal{O}$ and $\mathcal{M}$ have the same entailments over S. Besides safety, reuse of modules requires two additional properties namely *coverage* and *independence*.

Definition 5. (Module coverage [6]) *Let S be a signature and $\mathcal{T}'$, $\mathcal{T}$ be ontologies with $\mathcal{T}' \subseteq \mathcal{T}$ such that $S \subseteq Sig(\mathcal{T}')$. Then, $\mathcal{T}'$ guarantees coverage of S if $\mathcal{T}'$ is a module for S in $\mathcal{T}$.*

Definition 6. (Module Independence [6]) *Given an ontology $\mathcal{T}$ and signatures S_1, S_2, we say that $\mathcal{T}$ guarantees module independence if, for all $\mathcal{T}_1$ with $Sig(\mathcal{T}) \cap Sig(\mathcal{T}_1) \subseteq S_1$, it holds that $\mathcal{T} \cup \mathcal{T}_1$ is safe for S_2.*

Unfortunately, deciding whether or not a set of axioms is a minimal module is computationally hard or even impossible for expressive DLs [2, 3]. However, if the minimality requirement is dropped, good sized approximations can be defined that are efficiently computable, as in the case of *syntactic locality*, which modules are extracted in polynomial time.

Algorithm 3 (Extract a locality module [2])

`Procedure extract-module(`$\mathcal{T}, S, x$`)`
`Inputs: Tbox` $\mathcal{T}$`; signature` S`;` $x \in \bot, \top$`; Output` x`-module` $\mathcal{M}$

1 : $\mathcal{M} := \emptyset; \mathcal{T}' = \mathcal{T}$;
2 : **repeat**
3 : $\quad change = false$
4 : $\quad$ **for each** $\alpha \in \mathcal{T}'$
5 : $\quad\quad$ **if** α not x-local w.r.t. $S \cup \text{Sig}(\mathcal{M})$ **then**
6 : $\quad\quad\quad \mathcal{M} = \mathcal{M} + \{\alpha\}$
7 : $\quad\quad\quad \mathcal{T}' = \mathcal{T}' \setminus \{\alpha\}$
8 : $\quad\quad\quad changed = true$
9 : **until** $changed = false$

Definition 7. (Syntactic locality [3]) *Let* $\mathbf{S}$ *be a signature and* $\mathcal{O}$ *a* $\mathcal{SROIQ}$ *ontology. An axiom* α *is* $\bot$*-local w.r.t.* S *(*$\top$*-local w.r.t* $\mathbf{S}$*) if* $\alpha \in Ax(\mathbf{S})$*, as defined in the grammar:*

$\bot$*-Locality*

$\mathbf{Ax}(\mathbf{S})$	$::=$	$C^{\bot} \sqsubseteq C \vert C \sqsubseteq C^{\top} \vert w^{\bot} \sqsubseteq R \vert Dis(S^{\bot}, S) \vert Dis(S, S^{\bot})$
$\mathbf{Con}^{\bot}(\mathbf{S})$	$::=$	$A^{\bot} \vert \neg C^{\top} \vert C^{\bot} \sqcap C \vert C \sqcap C^{\bot} \vert C_1^{\bot} \sqcup C_2^{\bot} \vert \exists R^{\bot}.C \vert \exists R.C^{\bot}$
		$\vert \exists R^{\bot}.Self \vert \geqslant nR^{\bot}.C \vert \geqslant nR.C^{\bot}$
$\mathbf{Con}^{\top}(\mathbf{S})$	$::=$	$\neg C^{\bot} \vert C_1^{\top} \sqcap C_2^{\top} \vert C^{\top} \sqcup C \vert C \sqcup C^{\top} \vert \forall R.C^{\top} \vert \leqslant nR.C^{\top}$
		$\vert \forall R^{\bot}.C \vert \leqslant nR^{\bot}.C$

$\top$*-Locality*

$\mathbf{Ax}(\mathbf{S})$	$::=$	$C^{\bot} \sqsubseteq C \vert C \sqsubseteq C^{\top} \vert w \sqsubseteq R^{\top}$
$\mathbf{Con}^{\bot}(\mathbf{S})$	$::=$	$\neg C^{\top} \vert C^{\bot} \sqcap C \vert C \sqcap C^{\bot} \vert C_1^{\bot} \sqcup C_2^{\bot} \vert \exists R.C^{\bot} \vert \geqslant nR.C^{\bot}$
		$\vert \forall R^{\top}.C^{\bot} \vert \leqslant nR^{\top}.C^{\bot}$
$\mathbf{Con}^{\top}(\mathbf{S})$	$::=$	$A^{\top} \vert \neg C^{\bot} \vert C_1^{\top} \sqcap C_2^{\top} \vert C^{\top} \sqcup C \vert C \sqcup C^{\top} \vert \forall R.C^{\top} \vert$
		$\exists R^{\top}.C^{\top} \vert \geqslant nR^{\top}.C^{\top} \vert \leqslant nR.C^{\top} \vert \forall R^{\bot}.C \vert \leqslant nR^{\bot}.C$

In the grammar, we have that $A^{\bot}, A^{\top} \notin \mathbf{S}$ *is an atomic concept,* $R^{\bot}, R^{\top}$ *(resp.* $S^{\bot}$*,*$S^{\top}$*) is either an atomic role (resp. a simple atomic role) not in* $\mathbf{S}$ *or the inverse of an atomic role (resp. of a simple atomic role) not in* $\mathbf{S}$*,* C *is any concept,* R *is any role,* S *is any simple role, and* $C^{\bot} \in \mathbf{Con}^{\bot}(\mathbf{S})$*,* $C^{\top} \in \mathbf{Con}^{\top}(\mathbf{S})$*. We also denote by* $w^{\bot}$ *a role chain* $w = R_1 \circ \ldots \circ R_n$ *such that for some* i *with* $1 \leq i \leq n$*, we have that* R_i *is (possibly inverse of) an atomic role not in* $\mathbf{S}$*. An ontology* $\mathcal{O}$ *is* $\bot$*-local (*$\top$*-local) w.r.t.* S *if* α *is* $\bot$*-local (*$\top$*-local) w.r.t.* S *for all* $\alpha \in \mathcal{O}$*.*

Algorithm 3 may be used to extract either $\top$- or $\bot$-locality modules. Alternating the algorithm between $\bot$- and $\top$-locality module extraction until a fixed-point is reached results in $\bot\top^*$ modules.

3 Normal form

In this section we will introduce a normal form for any $\mathcal{SROIQ}$ ontology. The normal form is required to facilitate the conversion process between a $\mathcal{SROIQ}$ ontology and a hypergraph.

Definition 8. *Given* $B_i \in N_C \setminus \{\bot\}$*,* $C_i \in N_C \setminus \{\top\}$*,* $D \in \{\exists R.B, \geq nR.B, \exists R.Self\}$*,* $R_i, S_i \in N_R$ *and* $n \geqslant 1$*, a* $\mathcal{SROIQ}$ *ontology* $\mathcal{O}$ *is in* **normal form** *if every axiom* $\alpha \in \mathcal{O}$ *is in one of the following forms:*

α_1*:* $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$ α_2*:* $\exists R.B_1 \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$

α_3*:* $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \exists R.B_{n+1}$ α_4*:* $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \exists R.Self$

α_5*:* $\exists R.Self \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$ α_6*:* $\geqslant nR.B_1 \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$

α_7*:* $B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \geqslant nR.B_{n+1}$ α_8*:* $R_1 \circ \ldots \circ R_n \sqsubseteq R_{n+1}$

α_9*:* $D_1 \sqsubseteq D_2$

In order to normalize a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ we repeatedly apply the normalization rules from Table 2. Each application of a rule rewrites an axiom into an equivalent normal form. Algorithm 4 illustrates the conversion process.

Algorithm 4 *Given any $\mathcal{SROIQ}$ axiom α:*

1. *Recursively apply rules NR7 - NR11 to eliminate all equivalences, universal restrictions, atmost restrictions and complex role fillers.*
2. *Given that $\alpha = (\alpha_L \sqsubseteq \alpha_R)$, recursively apply the following steps until α_L contains no disjunctions and α_R contains no conjunctions:*
   (a) recursively apply rules NR1, NR3, NR6 to α_L,
   (b) recursively apply rules NR2, NR4, NR5 to α_R.
3. *recursively apply any applicable rules from NR12 through NR21.*

Table 2. $\mathcal{SROIQ}$ normalization rules

NR1	$\neg\hat{C}_2 \sqsubseteq \hat{C}_1 \rightsquigarrow \top \sqsubseteq \hat{C}_1 \sqcup \hat{C}_2$
NR2	$\hat{B}_1 \sqsubseteq \neg\hat{B}_2 \rightsquigarrow \hat{B}_1 \sqcap \hat{B}_2 \sqsubseteq \bot$
NR3	$\hat{B} \sqcap \hat{D} \sqsubseteq \hat{C} \rightsquigarrow \hat{B} \sqcap A \sqsubseteq \hat{C}$, $\hat{D} \sqsubseteq A$, $A \sqsubseteq \hat{D}$
NR4	$\hat{B} \sqsubseteq \hat{C} \sqcup \hat{D} \rightsquigarrow \hat{B} \sqsubseteq \hat{C} \sqcup A$, $\hat{D} \sqsubseteq A$, $A \sqsubseteq \hat{D}$
NR5	$\hat{B} \sqsubseteq \hat{C}_1 \sqcap \hat{C}_2 \rightsquigarrow \hat{B} \sqsubseteq \hat{C}_1$, $\hat{B} \sqsubseteq \hat{C}_2$
NR6	$\hat{B}_1 \sqcup \hat{B}_2 \sqsubseteq \hat{C} \rightsquigarrow \hat{B}_1 \sqsubseteq \hat{C}$, $\hat{B}_2 \sqsubseteq \hat{C}$
NR7	$\ldots \forall R.\hat{C} \ldots \rightsquigarrow \ldots \neg\exists R.A \ldots$, $A \sqcap \hat{C} \sqsubseteq \bot$, $\top \sqsubseteq A \sqcup \hat{C}$
NR8	$\ldots \exists R.\hat{D} \ldots \rightsquigarrow \ldots \exists R.A \ldots$, $\hat{D} \sqsubseteq A$, $A \sqsubseteq \hat{D}$
NR9	$\ldots \geqslant nR.\hat{D} \ldots \rightsquigarrow \ldots \geqslant nR.A \ldots$, $\hat{D} \sqsubseteq A$, $A \sqsubseteq \hat{D}$
NR10	$\ldots \leqslant nR.\hat{C} \ldots \rightsquigarrow \ldots \neg(\geqslant (n+1)R.\hat{C}) \ldots$
NR11	$\hat{B} \equiv \hat{C} \rightsquigarrow \hat{B} \sqsubseteq \hat{C}$,$\hat{C} \sqsubseteq \hat{B}$
NR12	$\geqslant 0R.B \sqsubseteq \hat{C} \rightsquigarrow \top \sqsubseteq \hat{C}$
NR13	$\hat{B} \sqsubseteq \exists R.\bot \rightsquigarrow \hat{B} \sqsubseteq \bot$
NR14	$\hat{B} \sqsubseteq \geqslant nR.\bot \rightsquigarrow \hat{B} \sqsubseteq \bot$
NR15	$\hat{B} \sqsubseteq \geqslant 0R.B \rightsquigarrow$
NR16	$\geqslant nR.\bot \sqsubseteq \hat{C} \rightsquigarrow$
NR17	$\exists R.\bot \sqsubseteq \hat{C} \rightsquigarrow$
NR18	$\hat{B} \sqcap \bot \sqsubseteq \hat{C} \rightsquigarrow$
NR19	$\bot \sqsubseteq \hat{C} \rightsquigarrow$
NR20	$\hat{B} \sqsubseteq \hat{C} \sqcup \top \rightsquigarrow$
NR21	$\hat{B} \sqsubseteq \top \rightsquigarrow$

Above $A \notin N_C$, $\hat{B}_i$ and $\hat{C}_i$ are possibly complex concept descriptions and $\hat{D}$ a complex concept description. $R \in N_R$, $n \geqslant 0$. We note that rules NR18 and NR20 makes use of the commutativity of $\sqcap$ and $\sqcup$.

Theorem 1. *Algorithm 4 converts any $\mathcal{SROIQ}$ ontology $\mathcal{O}$ to an ontology $\mathcal{O}'$ in normal form, such that $\mathcal{O}'$ is a conservative extension of $\mathcal{O}$. The algorithm terminates in linear time and adds at most a linear number of axioms to $\mathcal{O}$.*

For every normalized ontology $\mathcal{O}'$ the definition of syntactic locality from Definition 7 may now be simplified to that of Definition 9. This is possible since for every axiom $\alpha = (\alpha_L \sqsubseteq \alpha_R) \in \mathcal{O}'$, $\bot$-locality of α is dependent solely on α_L and $\top$-locality is dependent solely on α_R.

Definition 9. (Normal form syntactic locality) *Let* $\mathbf{S}$ *be a signature and* $\mathcal{O}$ *a normalized* $\mathcal{SROIQ}$ *ontology. Any axiom* α *is* $\bot$*-local w.r.t.* $\mathbf{S}$ *(*$\top$*-local w.r.t* $\mathbf{S}$*) if* $\alpha \in Ax(\mathbf{S})$*, as defined in the grammar:*

$\bot$-*Locality*	
$\mathbf{Ax}(\mathbf{S})$	$::= C^{\bot} \sqsubseteq C \mid w^{\bot} \sqsubseteq R \mid Dis(S^{\bot}, S) \mid Dis(S, S^{\bot})$
$\mathbf{Con}^{\bot}(\mathbf{S})$	$::= A^{\bot} \mid C^{\bot} \sqcap \mid C \sqcap C^{\bot} \mid \exists R^{\bot}.C \mid \exists R.C^{\bot} \mid \exists R^{\bot}.Self \mid \geqslant nR^{\bot}.C \mid \geqslant nR.C^{\bot}$
$\top$-*Locality*	
$\mathbf{Ax}(\mathbf{S})$	$::= C \sqsubseteq C^{\top} \mid w \sqsubseteq R^{\top}$
$\mathbf{Con}^{\top}(\mathbf{S})$	$::= A^{\top} \mid C^{\top} \sqcup C \mid C \sqcup C^{\top} \mid \exists R^{\top}.C^{\top} \mid \geqslant nR^{\top}.C^{\top} \mid \exists R^{\top}.Self$

In the grammar, we have that $A^{\bot}, A^{\top} \notin \mathbf{S}$ *is an atomic concept,* $R^{\bot}$,$R^{\top}$ *(resp.* $S^{\bot}$,$S^{\top}$*) is either an atomic role (resp. a simple atomic role) not in* $\mathbf{S}$ *or the inverse of an atomic role (resp. of a simple atomic role) not in* $\mathbf{S}$*,* C *is any concept,* R *is any role,* S *is any simple role, and* $C^{\bot} \in \mathbf{Con}^{\bot}(\mathbf{S})$*,* $C^{\top} \in \mathbf{Con}^{\top}(\mathbf{S})$*. We also denote by* $w^{\bot}$ *a role chain* $w = R_1 \circ \ldots \circ R_n$ *such that for some* i *with* $1 \leq i \leq n$*, we have that* R_i *is (possibly inverse of) an atomic role not in* $\mathbf{S}$*. An ontology* $\mathcal{O}$ *is* $\bot$*-local(*$\top$*-local) w.r.t.* $\mathbf{S}$ *if* α *is* $\bot$*-local(*$\top$*-local) w.r.t.* $\mathbf{S}$ *for all* $\alpha \in \mathcal{O}$*.*

We note that we may denormalize a normalized ontology if we maintain a possibly many-to-many mapping from normalized axioms to their original source axioms. Formally, define a function $denorm : \hat{\mathcal{O}} \rightarrow 2^{\mathcal{O}}$, with $\mathcal{O}$ an $\mathcal{SROIQ}$ ontology and $\hat{\mathcal{O}}$ its normal form. For brevity, we write $denorm(\Phi)$, with Φ a set of normalized axioms, to denote $\bigcup_{\alpha \in \Phi} denorm(\alpha)$.

4 $\mathcal{SROIQ}$ hypergraph

Suntisrivaraporn [11] showed that for the DL $\mathcal{EL}^{+}$, extracting $\bot$-locality modules are equivalent to the reachability problem in directed hypergraphs. This was extended in [9, 10] to include a reachability algorithm for $\top$-locality modules. In this section we show that a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ in normal form can be mapped to a hypergraph which preserves both $\bot$-locality and $\top$-locality.

Definition 10. *Let* α *be a normalized axiom and* $\alpha_{\bot}$ *a minimum set of symbols from Sig(*α*) required to ensure that* α *is not* $\bot$*-local, and let* $\mathcal{H} = (\mathcal{V}, \mathcal{E})$ *be a hypergraph. We say that an edge* $e \in \mathcal{E}$ *preserves* $\bot$*-locality iff* $\alpha_{\bot} = T(e)$*. Similarly,* $e \in \mathcal{E}$ *preserves* $\top$*-locality whenever* $\alpha_{\top} = H(e)$*.*

For each normal form axiom α_i in Definition 8 we show that α_i may be mapped to a set of hyperedges, with nodes denoting symbols from $\mathrm{Sig}(\alpha_i)$, such that both $\bot$-locality and $\top$-locality are simultaneously preserved.

- Given $\alpha_1 : B_1 \sqcap \ldots \sqcap B_n \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$ we map it to the hyperedge $e_{\alpha_1} = (\{B_1, \ldots, B_n\}, \{C_1, \ldots, C_m\})$. We transform the hyperedge e_{α_1} to two new hyperedges $e^{B}_{\alpha_1} = (\{B_1, \ldots, B_n\}, \{H_1\})$ a B-hyperedge, $e^{F}_{\alpha_1} =$

$(\{H_1\}, \{C_1, \ldots, C_m\})$ an F-hyperedge and with H_1 a new node. By definition each C_j is B-connected to H_1 if all B_i are B-connected to H_1. From Definition 9 we know that this preserves $\bot$-locality for α_1 since it is $\bot$-local, w.r.t. a signature S, exactly when any of the conjuncts $B_i \notin S$. In other words it is non $\bot$-local exactly when all $B_i \in S$. The same also holds for $\top$-locality, since $e^F_{\alpha_1}$ requires every $C_i \in \alpha_1$ to be in S for H_1 to be F-connected. From Definition 9 we see that, w.r.t. a signature S, $e^F_{\alpha_1}$ is $\top$-local exactly when any of the disjuncts $C_i \notin S$.

- Given $\alpha_2 : \exists R.B_1 \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$ or $\alpha_6 : \geqslant nR.B_1 \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$ we map it to the two hyperedges $e^B_{\alpha_{2/6}} = (\{B_1, R\}, \{H_2\})$, $e^F_{\alpha_{2/6}} = (\{H_2\}, \{C_1, \ldots, C_m\})$ an F-hyperedge and with H_2 a new node. This mapping preserves $\bot$-locality for $\alpha_{2/6}$ since by Definition 9 it is $\bot$-local, w.r.t. a signature S, exactly when either B_1 or R is not in S. The argument for $\top$-locality follows that of α_1.
- Given $\alpha_3 : B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \exists R.B_{n+1}$ or $\alpha_7 : B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \geqslant nR.B_{n+1}$ we map it to the hyperedges $e^B_{\alpha_{3/7}} = (\{B_1, B_2, \ldots, B_{n-1}, B_n\}, \{H_3\})$, $e^{F_1}_{\alpha_{3/7}} = (\{H_3\}, \{B_{n+1}\})$, $e^{F_2}_{\alpha_{3/7}} = (\{H_3\}, \{R\})$. This mapping preserves $\bot$-locality for $\alpha_{3/7}$ similarly to $e^B_{\alpha_1}$ for α_1. From Definition 9 we know that $\top$-locality for either of these axioms, w.r.t. a signature S, is dependent on neither R nor B_{n+1} being elements of S. Therefore, they are non $\top$-local exactly when either or both of these are in S. This is represented by the two edges $e^{F_1}_{\alpha_{3/7}}$ and $e^{F_2}_{\alpha_{3/7}}$ for which H_3 becomes F-connected exactly when either R or B_{n+1} is F-connected.
- Given $\alpha_4 : B_1 \sqcap \ldots \sqcap B_n \sqsubseteq \exists R.Self$ and $\alpha_5 : \exists R.Self \sqsubseteq C_1 \sqcup \ldots \sqcup C_m$ we see that $\exists R.Self$ is both $\bot$ or $\top$ local exactly when $R \notin S$. Therefore we map α_4 to the hyperedge $e^B_{\alpha_4} = (\{R\}, \{C_1, \ldots, C_m\})$, and α_5 to the hyperedge $e^F_{\alpha_5} = (\{B_1, \ldots, B_n\}, \{R\})$.
- Given $\alpha_8 : R_1 \circ \ldots \circ R_n \sqsubseteq R_{n+1}$, we see that α_8 is $\bot$-local exactly when any $R_i \notin S, i \leq n$ and is $\top$-local exactly when $R_{n+1} \notin S$. We therefore map α_8 to the hyperedge $e^B_{\alpha_8} = (\{R_1, \ldots, R_n\}, \{R_{n+1}\})$.
- For α_9 we have many forms, all variants of those discussed in the previous mappings. Therefore α_9 is mapped to any of the following: $e^{B_1}_{\alpha_9} = (\{R, B_1\}, \{H_9\}\})$, $e^{F_1}_{\alpha_9} = (\{H_9\}, \{R\})$, $e^{F_2}_{\alpha_9} = (\{H_9\}, \{B\})$, or $e^1_{\alpha_9} = (\{R, B_1\}, \{R\})$, or $e^{F_1}_{\alpha_9} = (\{R_1\}, \{R_2\})$, $e^{F_2}_{\alpha_9} = (\{R_1\}, \{B\})$, or $e^1_{\alpha_9} = (\{R_1\}, \{R_2\})$.

Given a $\mathcal{SROIQ}$ ontology $\mathcal{O}$ in normal form we may now map every axiom $\alpha \in \mathcal{O}$ to its equivalent set of hyperedges. For each of these mappings there are at most three hyperedges introduced, therefore mapping the whole ontology $\mathcal{O}$ to an equivalent hypergraph $\mathcal{H}_\mathcal{O}$ will result in a hypergraph with the number of edges at most linear in the number of axioms in $\mathcal{O}$. It is easy to show that the mapping process can be completed in linear time in the number of axioms in $\mathcal{O}$.

We note that, similar to the normalization process, we may maintain a possibly many-to-many mapping from normalized axioms to their associated hyperedges. Formally, define a function $deedge : \mathcal{H}_\mathcal{O} \to 2^\mathcal{O}$, with $\mathcal{O}$ a $\mathcal{SROIQ}$ ontology and $\mathcal{H}_\mathcal{O}$ its hypergraph. For brevity, we write $deedge(\Phi)$, with Φ a set of hyperedges, to denote $\bigcup_{e \in \Phi} deedge(e)$.

5 Hypergraph module extraction

In this section we show that, given a hypergraph $\mathcal{H}_\mathcal{O}$ for a $\mathcal{SROIQ}$ ontology $\mathcal{O}$, we may extract a frontier graph from $\mathcal{H}_\mathcal{O}$ which is a subset of a $\bot\top^*$ module. We show that some of these modules guarantee safety, module coverage and module independence. The hypergraph algorithms presented require one start node s and a destination node t. In order to extend these algorithms to work with an arbitrary signature S, we introduce a new node s with with an edge $e_{s_i} = (s, s_i)$ for each $s_i \in S \cup \top$, as well as a new node t with an edge $e_{t_i} = (s_i, t)$ for each $s_i \in S \cup \bot$.

Theorem 2. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology and $\mathcal{H}_\mathcal{O}$ its associated hypergraph and S a signature. Algorithm 1 - Bvisit extracts a set of B-hyperpaths $\mathcal{H}^B_\mathcal{O}$ corresponding to the $\bot$-locality module for S in $\mathcal{O}$. Therefore, these modules also guarantees safety, module coverage and module independence.*

Theorem 3. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology and $\mathcal{H}_\mathcal{O}$ its associated hypergraph and S a signature. Algorithm 1 - Fvisit extracts a set of F-hyperpaths $\mathcal{H}^F_\mathcal{O}$ corresponding to a subset of the $\top$-locality module for S in $\mathcal{O}$.*

Theorem 4. *Let $\mathcal{O}$ be a $\mathcal{SROIQ}$ ontology and $\mathcal{H}_\mathcal{O}$ its associated hypergraph and S a signature. Algorithm 2 extracts a frontier graph $\mathcal{H}^{BF}_\mathcal{O}$ corresponding to a subset of the $\bot\top^*$-locality module for S in $\mathcal{O}$.*

The module extracted in Theorem 3 is a subset of the $\top$-locality module for a given seed signature. It is as yet unclear whether or not these modules provide all the model-theoretic properties associated with $\top$-locality modules. However, from the previous work done for the DL $\mathcal{EL}^+$ [10], it is evident that these modules preserve all entailments for a given seed signature S. Further, they also preserve and contain all justifications for any given entailment. Similarly, the exact module theoretic properties of modules associated with frontier graphs is something we are currently looking into.

6 Conclusion

We have introduced a normal form for any $\mathcal{SROIQ}$ ontology, as well as the necessary algorithms in order to map any $\mathcal{SROIQ}$ ontology to a syntactic locality preserving hypergraph. This mapping process can be accomplished in linear time in the number of axioms with at most a linear increase in the number of hyperedges in the hypergraph.

Standard path searching algorithms for hypergraphs may now be used to find: (1) sets of B-hyperpaths — this is equivalent to finding $\bot$-syntactical locality modules; (2) sets of F-hyperpaths — these are subsets of $\top$-locality modules, and (3) frontier graphs — these are subsets of $\bot\top^*$ modules. Whilst the modules associated with B-hyperpaths share all the module theoretic properties of $\bot$-locality modules, it is unclear at this point which module-theoretic properties modules associated with F-hyperpaths and frontier graphs possess.

The ability to map $\mathcal{SROIQ}$ ontologies to hypergraphs, such that hyperedges preserve syntactic locality conditions, allows us to investigate the relationship between DL reasoning algorithms and the vast body of standard hypergraph algorithms in greater depth.

Our primary focus for future research is to investigate and define the module-theoretic properties of modules associated with F-hyperpaths and frontier graphs as well as their relative performance with respect to existing locality methods. Thereafter, we aim to expand our research and investigate other hypergraph algorithms and how they may be applied to DL reasoning problems.

References

1. Cuenca Grau, B., Halaschek-Wiener, C., Kazakov, Y., Suntisrivaraporn, B.: Incremental classification of description logic ontologies. Tech. rep. (2012)
2. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Just the right amount: extracting modules from ontologies. In: Williamson, C., Zurko, M. (eds.) Proceedings of the 16th International Conference on World Wide Web (WWW '07). pp. 717–726. ACM, New York, NY, USA (2007)
3. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. Journal of Artificial Intelligence Research (JAIR) 31, 273–318 (2008)
4. Del Vescovo, C., Parsia, B., Sattler, U., Schneider, T.: The modular structure of an ontology: atomic decomposition and module count. In: O. Kutz, T.S. (ed.) Proc. of WoMO-11. Frontiers in AI and Appl., vol. 230, pp. 25–39. IOS Press (2011)
5. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistable $\mathcal{SROIQ}$. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) Proceedings of the Tenth International Conference on Princleples of Knowledge Representation and Reasoning. pp. 57–67. AAAI Press (2006)
6. Jiménez-Ruiz, E., Cuenca Grau, B., Sattler, U., Schneider, T., Berlanga, R.: Safe and economic re-use of ontologies: A logic-based methodology and tool support. In: Proceedings of ESWC-08. vol. 5021 of LNCS, pp. 185–199 (2008)
7. Maier, F., Ma, Y., Hitzler, P.: Paraconsistent OWL and related logics. In: Janowicz, K. (ed.) Semantic Web 2012. pp. 1–33. IOS Press (2012)
8. Nguyen, S., Pretolani, D., Markenzon, L.: On some path problems on oriented hypergraphs. Theoretical Informatics and Applications 32(1-2-3), 1–20 (1998)
9. Nortjé, R.: Module extraction for inexpressive description logics. Master's thesis, University of South Africa (2011)
10. Nortjé, R., Britz, K., Meyer, T.: Bidirectional reachability-based modules. In: Proceedings of the 2011 International Workshop on Description Logics (DL2011). CEUR Workshop Proceedings, CEUR-WS (2011), http://ceur-ws.org/
11. Suntisrivaraporn, B.: Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies. Ph.D. thesis, Technical University of Dresden (2009)
12. Thakur, M., Tripathi, R.: Complexity of linear connectivity problems in directed hypergraphs. Linear Connectivity Conference pp. 1–12 (2001)

Two Case Studies of Ontology Validation

Doug Foxvog

University of Maryland Baltimore County, Baltimore, MD, USA
doug@foxvog.org

Abstract. As ontologies and ontology repositories have proliferated, the need for a discipline of ontology validation and quality assurance has grown more urgent. This report describes two case studies of ontology validation by converting ontologies to a more powerful reasoning language and analyzing them using logical queries. The lessons learned and directions for continuing work are discussed.

Keywords: ontology, quality assurance, validation

1 Introduction

In computer science, an ontology is a formalization of the concepts of a specific field, specifying the types (classes) of things that are dealt with in the field, relations that may apply among instances of those classes, rules applying to instances of those classes, and possibly specific instances of those classes. It may define subsumption and disjointness between classes or relations and may constrain the argument types of relations. An ontology is not a definition of terms in a natural language, although many ontologies provide mappings between the terms of the ontology and natural language terms.

If an ontology accurately constrains the relations and classes of a model of the domain with restrictions that prevent assertions that could not be true in the modeled domain and does so in a logically consistent manner, then it can be used to encode valid information in the field, conclude additional information that is implied by the stated information, and detect or block statements that are inconsistent with the domain model.

However, an ontology that does not accurately model the domain would allow logically invalid statements to be asserted, prevent true statements from being made, or both. An ontology may be incorrect not only due to some of its statements being incorrect, but also due to missing assertions. An ontology that accurately encodes a domain model and yet is logically invalid indicates that the model itself is invalid.
For these reasons, it is important to validate ontologies before use and whenever they are modified. Not only can sets of logically inconsistent statements be identified, but omission of argument constraints and class disjointness assertions can be flagged.

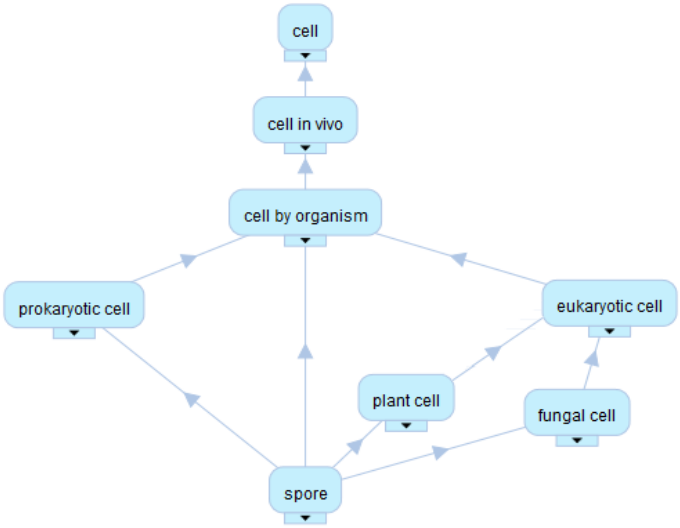

Fig. 1. Class with three disjoint superclasses[1]

Our method does not cover ways to verify if an ontology corresponds to reality or to an external model, but deals with design flaws and logical issues that could be examined with an inference engine. This paper presents the results of validating two standard ontologies with strong user bases and communities (the Cell Line Ontology and Plant Ontology described in Section 2) using this technique.

As the ontology language for the selected case studies lacks the reasoning capabilities for logically detecting all the considered types of ontology flaws, we translated the ontologies into a richer language (CycL [1]) in order to perform the analysis[2]. Some errors are flagged as the translated ontology is being input to the Cyc system, while others can only be detected by asking the inference engine queries about the ontology.

2 Source of Ontologies for Validation

The National Center for Biomedical Ontology maintains hundreds of ontologies for the biomedical field with versions in several formats [2].

We selected as case studies two associated ontologies hosted by the NCBO: the Cell Line Ontology and the Plant Ontology, downloading them in the OBO format [3]. The 5 May 2009 version of the Cell Line Ontology [4] included thousands of cell types including types of animal cells, plant cells, fungal cells, and prokaryotic cells.

[1] http://bioportal.bioontology.org/ontologies/39927/?p=terms&conceptid=CL%3A0000522

[2] Although CycL is formally an undecidable language, the queries used here (taxonomic, local closed world, queries ranging over locally defined predicates, etc.) were not. The Cyc inference engine specifies whether lists of answers provided are known to be complete.

The Plant Ontology [5] covers plant anatomy (including types of plant cells), morphology, growth, and development. The Cell Line Ontology and Plant Ontology had hundreds of pairs of plant cell concepts, with the same name and same or similar English definition, but different IDs.

Disjointness violations we detected in the Cell Line Ontology (see 3.1) suggested that at least the one ontology had not been created with automated logical verification of its statements. We decided to do a more complete analysis of the two ontologies to determine if there were additional issues.

3 Cell Line Ontology

3.1 Introduction

In the Cell Line Ontology terms for several classes of cells, such as "epidermal cell," which are used by botanists and anatomists to refer to cells with similar functions in both plants and animals, had been created with some plant cell subclasses and other animal cell subclasses. However, at some point these terms were defined as subclasses of animal cell. Without disjointness constraints between plant and animal cell, this situation was not detected when the statements were made. The term for "spore" was similarly a subclass of three disjoint classes: "prokaryotic cell," "plant cell," and "fungal cell" (Fig. 1) and "zygote" was a subclass of "animal cell" and "plant cell."

These disjointness issues, which were detected by Cyc when we attempted to add disjointness assertions to a translated version of the 2009 Cell Line Ontology, were corrected with the separation of plant cell types from the Cell Line Ontology in December 2011 [6].

3.2 Analysis

For a more complete analysis, we downloaded an updated (13/1/2012 09:59) version of the Cell Line Ontology. This version had obsoleted all plant cell types, referring the user to the Plant Ontology for such terms, and distinguished prokaryotic spores from fungal spores. The ontology defines 1928 non-obsoleted cell types, 29 binary predicates, and 32 (new) disjointness assertions among cell types.

A collection of terms from other ontologies (including PR for protein, UBERON - cross-species anatomy, NCBI - biological taxa, ChEBI - chemical entities, PATO-phenotypic qualities) are also included to be specified as arguments to relations restricting the cell type definitions. 4233 such assertions are included in the ontology.

To perform an analysis, the ontology was converted to CycL, loaded into OpenCyc [7], and then queries were asked using the OpenCyc interface.

Formal criteria Analysis of the logical constraints for the Cell Line Ontology showed that the cell types were arranged in a directed acyclic graph rooted on a term for "cell" and that there were no shared subclasses of any of the defined disjoint pairs

(Table 1, column 1). Cyc was not needed for such a determination – OWL reasoners can detect intersections of disjoint classes.

Table 1. Queries of Cell Line Ontology

Disjoint classes that have a common subclass	Cell types that develop from Eukaryotic cells, but are not known to be Eukaryotic	Eukaryotic cell types that develop from cell types not known to be Eukaryotic
(and (ist-Asserted[3] CL_Mt (disjointWith ?C1 ?C2)) (genls ?C0 ?C1) (genls ?C0 ?C2))	(and (allRelationExists ?C1 CL_developsFrom ?C2) (genls ?C2 EukayoticCell) (unknownSentence (genls ?C1 EukayoticCell)))	(and (allRelationExists ?C1 CL_developsFrom ?C2) (genls ?C1 EukayoticCell) (unknownSentence (genls ?C2 EukayoticCell)))
Answers: 0	Answers: 19	Answers: 22

Informal Criteria – Completeness. Nine of the 29 binary relations had argument restrictions defined, all of which were to the PATO Ontology's term for "Quality" (PATO:00000001). Five of these relations were defined as transitive, two of them having an identical domain and range defined, and the rest having neither. These relations were only used in expressing intersection with a property [See Table 2], and in all cases the classes were consistent with the argument restrictions. The lack of argument restrictions on most relations is a significant incompleteness.

One of the properties defined for many cell types is that they develop from other cell types. Logically, cells that develop from types of EukayoticCell[4] (or Prokaryotic_Cell or Animal_Cell) should themselves be types of EukayoticCell (or Prokaryotic_Cell or Animal_Cell). The inference engine finds 19 violations of this principle. Similarly, if a subtype of one of these general classes of cells is known to develop from another type, it is quite possible that the second type is also a subtype of the general class. The inference engine finds 22 cases in which the cell type from which a eukaryotic cell type develops is not known to be a eukaryotic cell type. Table 1 (columns 2 and 3) provides the queries asked of the inference engine, their English translations, and the number of answers.

[3] The CycL relation ist-Asserted relates a specified context to a specific statement made in it; the relation genls is the CycL subclass relation; allRelationExists means that for every instance of a class (first argument) the specified relation (second arg.) relates it to some instance of another class (third arg.); unknownSentence means that the statement that is its only argument is neither stated nor derivable through taxonomic reasoning. Variables in CycL are prefixed by a question mark ("?"). The relation ist-Asserted can not be expressed in FOL.

[4] The Cell Ontology uses IDs such as CL:0000003. For clarity, we use the phrase provided by the name field to specify each term.

Table 2. Germ line stem cell defined as intersection of germ-line cell and being capable of stem cell division (OWL format)

```
:CL_0000014 rdf:type owl:Class ;
            owl:equivalentClass
             [ rdf:type owl:Class ;
               owl:intersectionOf
               ( :CL_0000039
                 [ rdf:type owl:Restriction ;
                   owl:onProperty : capable_of ;
                   owl:someValuesFrom GO:0017145
                 ]
               )
             ]
```

Only 32 disjointness assertions are defined, all of which apply to types of white blood cells and blood progenitor cells. Cell types near the top of the hierarchy include cell types by number of nuclei (none, some, one, greater than one) and cell types by organism type (prokaryotic, eukaryotic, animal, and fungal — plant cells having been removed from the ontology), which strongly indicated missing partitions and disjointness assertions (Fig. 2).

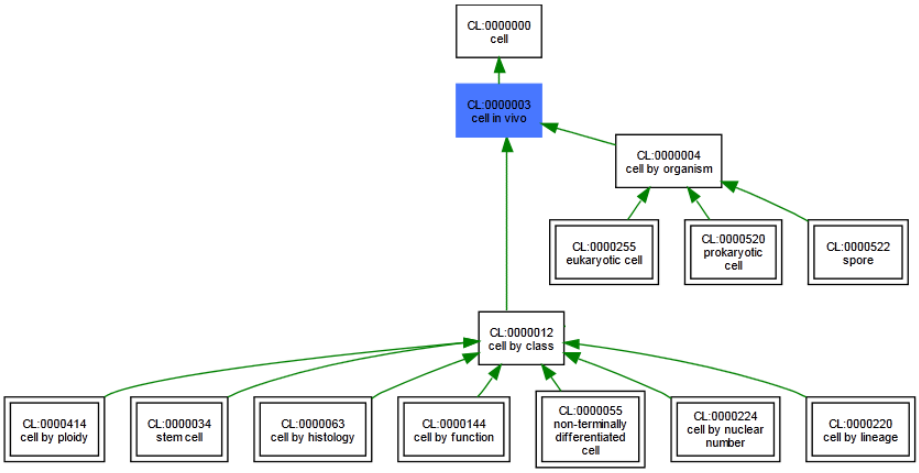

Fig. 2. Top Layers of Cell Ontology hierarchy, from http://proto.informatics.jax.org/prototypes/GOgraphEX/CL_Graphs/CL_0000003.svg

Informal Criteria – Abstraction Level. A brief analysis of the very top levels of the hierarchy showed that sets of classes were being treated as normal classes, with their members being labeled as subclasses. The initial partition (as described by textual descriptions) is `Cell_in_Vitro` vs. `Cell_in_Vivo` (renamed `Native_Cell`) with almost every other cell type a subclass of `Cell_in_Vivo`.

An ontologist might prefer to make this distinction orthogonal to other distinctions (since *in vitro* cells might reasonably be considered to be muscle/nerve/etc. cells, and

to have a nucleus or not even though they are not in a body). Cell_in_Vivo has the subclasses Cell_by_Organism and Cell_by_Class. Cell_by_Class has the subclasses Cell_by_Nuclear_Number, Cell_by_Ploidy, Cell_by_Lineage, Cell_by_Histology, Cell_by_Function, and Nonterminally_Differentiated_Cell. The textual descriptions of each of the "Cell_by_X" classes start with "a classification of cells by …," making it clear that each are intended to be sets of classes, i.e. metaclasses, since their descriptions are not generally applicable to the various subclasses of those cell types that are defined as their direct subclasses. The definitions of these classes are meaningless with respect to the individual cells that are supposed to be their instances [8].

Some of these sets of cell types seem to naturally be disjoint sets. Under Cell_by_Organism there is Prokaryotic_Cell and Eukaryotic_Cell and under Eukaryotic_Cell there is Animal_Cell, Fungal_Cell, and Mycetozoan_Cell (Plant_Cell has been obsoleted), all of which are cells distinguished by the type of organism of which they are a part. Since every organism is either a prokaryote or a eukaryote, the first division is a partition on Cell although it has not been so declared in the ontology. The three directly specified subclasses of Eukaryotic Cell are all disjoint, but this is not stated in the ontology; these subclasses do not cover all eukaryotic cells, so it is not a partition.

A similar analysis covers Cell_by_Nuclear_Number and Cell_by_Ploidy, each of which has instances (although defined as subclasses) that partition Cell. These instances each have very few subclasses even though most cell types fall under the definition of an instance of each of these metaclasses.

Table 3. Cell Line Ontology Property Issues

Cell types defined as the intersection of another cell type and having some property	Cell types defined as the intersection of a metatype and having some property	Cell types that have a property and are not stated as being a subclass of the intersection of a superclass with the property
(and (isIntersectionOf ?C ?C1 ?PRED ?V) (isa ?C1 CellType) (isa ?C CellType))	(and (isIntersectionOf ?CT ?MCT ?PRED ?VALUE) (genls ?MCT CellType) (genls ?CT Cell))	(and (isIntersectionOf ?C1 ?C0 ?PRED ?V) (genls ?C2 ?C0) (allRelationExists ?C2 ?PRED ?V) (unknownSentence (genls ?C2 ?C1)))
Answers: 547	Answers: 10	Answers: 10

Formal criteria – Internal Consistency. Over 4200 cell types in the ontology are defined as having some property (e.g., haploid, mononucleate, etc.). Over 500 cell types are defined as being the intersection of a more general cell type and having a specific property. Ten of the cell types which end up being instances of one of the metaclasses are also defined as being an intersection of a metaclass and having one of

these properties. However, in the Cell Line Ontology, the more specific cell types specified as having a property are not always (through the subclass hierarchy) declared to be subclasses of the class which is an intersection of that property and one of their superclasses. Although many reasoners (including OWL reasoners) can derive the subclass relationship, BioPortal's browser for the Cell Type Ontology at the time of the ontology release did not conclude them.

A query by the inference engine yielded ten such missing subclass relationships, which makes the ontology internally inconsistent for insufficiently powerful systems such as the BioPortal ontology browser of January 2011. Table 3 provides queries asked of the inference engine, their English translations, and the number of answers.

3.3 Resolution

We converted the metaclasses at the top of the ontology to actual metaclasses, converting the subclass assertions of their instances to instantiation assertions. This meant that those classes which had their `subclassOf` relationships removed needed to have new `subclassOf` relationships asserted if no others existed. Subclass assertions were made from the instances of these metaclasses to `Cell`, not to `Native_Cell`. `Prokaryotic_Cell` was made a subtype of `Anucleate_Cell` and `Nucleate_Cell` was made a subclass of `Eukaryotic_Cell`. Other subclass assertions are needed at this level; for example, a number of the (now) instances of `Cell_Type_by_Function` should be declared to be subclasses of `Animal_Cell` or `Eukaryotic_Cell`, but such work is the responsibility of a developer or subject matter expert.

Defined subclasses of these now direct instances of the metaclasses were examined to determine whether they should also be instances of the metaclass and were so asserted only if judged appropriate. For example, `Cell_By_Nuclear_Number` had `Mononucleate_Cell`, `Binucleate_Cell`, and `Multinucleate_Cell` added as instances while remaining as subclasses of `Nucleate_Cell`.

Other former direct subclasses were examined to determine whether they should be subclasses of direct instances of the metaclass, and not instances themselves. For example, `Cell_by_Nuclear_Number` had its instances restricted to `Anucleate_Cell`, `Nucleate_Cell`, `Mononucleate_Cell`, and `Multinucleate Cell`, with its other former direct subclasses (`Mononuclear_Osteoclast`, `Multinuclear_Osteoclast`, …) being asserted as subclasses of the appropriate direct instance as indicated by their comments.

Disjointness statements were made for the instances of the newly restructured metaclasses, `Cell_by_Organism`, `Cell_by_Nuclear_Number`, and `Cell_by_Ploidy`. `Cell_by_Organism` was made a subclass of `Cell_by_Class`.

We added rules to the CycL version of the ontology conclude subclass relationships:

- A rule was added so that cell types that are defined as developing from eukaryotic or animal cell types are concluded to also be subclasses of `Eukaryotic_Cell`

or Animal_Cell, respectively. This resulted in 26 subclass assertions being derived.

- A rule was added so that if one class is defined as an intersection of a class and a property, subclasses of that class that have that property are concluded to be subclasses of the intersection class. This resulted in a further ten subclass assertions being derived.
- A rule was added so that if one class is defined as an intersection of a metaclass and a property, other classes with that property are concluded to be subclasses of the direct instance of the metaclass. The intersection assertion was changed to being an intersection of Cell and the property. This resulted in nine subclass assertions being derived.

The Cell Ontology obsoleted the metaclasses in March 2012 [8]. A more recent OBO Library browser does conclude subclass relationships derived from intersection definitions.

Other detected problems still need to be resolved. Such work is not the responsibility of a validator, but of a developer or subject matter expert. We recommend that Cell Line Ontology developers:

- Define subclasses of Mononucleate_Cell and other instances of Cell_by Nuclear_Number so that every cell type that has a restricted nuclear number is defined as such by the subclass hierarchy.
- Define subclasses of Diploid_Cell and other instances of Cell_by_Ploidy so that every cell type that has a restricted ploidy is defined as such by the subclass hierarchy.
- Define those instances of Cell_by_Function which of necessity are subclasses of Animal_Cell or Eukaryotic_Cell as being so. For those instances which are not so restricted, check their direct subclasses to determine whether they should be subclasses of Animal_Cell or Eukaryotic_Cell.
- In cases in which a subclass of Eukaryotic_Cell (or Animal_Cell) is declared to develop from a cell type that is not such a subclass, the second class should be examined to determine whether it should be a subclass of Animal_ Cell or Eukaryotic_Cell.
- Add many more disjointness assertions among sibling classes, as appropriate.
- Define appropriate argument restrictions on the predicates in the ontology.

4 Plant Ontology

4.1 Introduction

The 2 April 2012 version of the Plant Ontology contains 1593 terms, 1181 of which are types of plant anatomical entity, 272 of which are types of plant structure developmental stage, eight of which are binary relations, and 132 of which are obsoleted. 37 disjointness assertions among cell types are included. The Plant Ontology includes

64 assertions specifying that one class is an intersection of another class with having a specific property.

The intersection assertions are accepted as a way of stating subclass relationships that are to be concluded instead of directly stated. This was done in order to avoid directly stating "dual parentage" in the ontology [5, p. 4].

4.2 Analysis

To analyze the ontology, it was converted to CycL, loaded into OpenCyc, and queries were asked using the OpenCyc interface.

Formal criteria – Logical constraints. Analysis of the logical constraints for the Plant Ontology showed that the classes were arranged in two directed acyclic graphs rooted on terms for "plant anatomical entity" and "plant structure development stage," and that there were no shared subclasses of any of the defined disjoint pairs. There was no violation of logical constraints.

Formal criteria – Internal Consistency. Over 800 classes in the ontology are defined as having some property. 64 of the classes are defined as being an intersection of a more general class and having one of these properties. By querying the inference engine, we found that in 63 cases, the more specific classes are not (directly or indirectly) defined as subclasses of the class-property intersection. Two examples of this are types of plant cell that have the property of being part of a plant embryo, but are not known to be subclasses of `EmbryonicPlantCell`. For systems with limited reasoning capabilities, these are violations of internal consistency.

Table 3 provides the queries asked of the inference engine, their English translations, and the number of answers.

Table 2. Plant Ontology Property Issues

Classes which are defined to have some property that are not defined to be subclasses of the intersection of a superclass with that property	Plant cell types that are part of plant embryos, but are not known to be embryonic plant cells

`(and`
` (isIntersectionOf`
`  ?P1 ?P0 ?PRED ?V)`
` (allRelationExists ?P2 ?PRED ?V)`
` (genls ?P2 ?P0)`
` (unknownSentence`
`  (genls ?P2 ?P1)))`
	`(and`
	`  (allRelationExists ?P1`
	`  PO_part_of PlantEmbryo)`
	`  (genls ?P1 PlantCell)`
	`  (unknownSentence`
	`  (genls ?P1`
	`  EmbryonicPlantCell)))`

`Answers: 63`	`Answers: 2`

Informal Criteria – Completeness. Disjointness assertions were missing from the developmental stage hierarchy and from near the top of the anatomical hierarchy. None of the binary relations had argument restrictions defined. Three of these relations were defined as transitive; none as symmetric or reflexive. Only 37 disjointness assertions are present, all of which are well down in the cell type hierarchy. There are

significant gaps in the ontology in both argument type restrictions and disjointness assertions.

Informal Criteria – Abstraction Level. Unlike the Cell Line Ontology, the Plant Ontology had no metaclasses near the top of the hierarchy that were used as subclasses.

4.3 Resolution

We added a rule to conclude subclass relationships:

- A rule was added so that if one class is defined as an intersection of a class and a property, subclasses of that class that have that property are concluded to be subclasses of the intersection class. This resulted in 63 assertions being derived.

A more recent Plant Ontology browser does conclude subclass relationships derived from intersection definitions. Much is still missing, e.g., disjointness assertions and argument type restrictions. Such work is not the responsibility of a validator, but of a developer or subject matter expert.

We recommend that Plant Ontology developers:

- Specify disjointness among sibling classes as appropriate.
- Define appropriate argument restrictions on the predicates in the ontology.
- Review comments which state that a class has a certain property, and encode those that are valid and are not already encoded or derivable from properties of superclasses.

5 Lessons Learned and Conclusion

We analyzed two ontologies that have strong user bases and communities for ensuring their validity. Significant problems were discovered with each ontology as a result of verification queries.

We note that public ontologies are not static. Early problems in the class hierarchy of the Cell Line Ontology, discovered when making high-level disjointness assertions (e.g., plant vs. animal cell) flagged common subclasses, were corrected before our in-depth analysis and the Plant Ontology was disconnected from the Cell Type Ontology in December of 2011. The in-depth analyses of the two ontologies discovered no remaining disjointness problems. Only a domain expert can determine whether this is due to the validity of the current subclass hierarchy or the sparseness of disjointness assertions.

One of the two ontologies erroneously treated metaclasses as normal subclasses of the root term. This led to numerous missing subclass assertions (evidently because the subclass does not fit the definition of the metaclass). These metaclasses have since been obsoleted. They could be reinstated as metaclasses if they are recognized as such.

The omission of argument restrictions can be readily determined. The lack of disjointness assertions among sibling classes can also be readily determined, but a subject matter expert should determine whether such sibling classes are actually disjoint.

Both ontologies had statements that instances of certain classes had certain properties, and that other classes were the intersection of superclasses with having some property. Such statements were initially not executable rules in the provided ontology viewer, so that in both cases subclass assertions that could be concluded based on these rules were missing. These examples emphasize that ontology evaluation needs to cover more than just whether the statements in an ontology lead to a logical contradiction.

When an ontology includes statements that could be mapped to rules from which subclass relationships or disjointness between classes can be concluded, an ontology evaluation step in a sufficiently rich semantic language can draw such conclusions and check if the conclusions are entailed by the encoded subclass and disjointness statements. If they are not already present, the concluded statements can then be added to the ontology.

The presence of metaclasses erroneously defined as normal classes in a subsumption hierarchy cannot be concluded from automatic analysis of the statements in an ontology. Such problems may be more likely to occur near the root of a subclass hierarchy and can be manually detected by reading the descriptions of the terms. Such situations can be resolved by determining which of the defined subclasses of the metaclass are normal classes and which are metaclasses, converting the normal classes to be instances instead of subclasses of the metaclass, and adding disjointness assertions, as appropriate, among them.

It is noteworthy that the problems found in these case studies consisted of systematic repetition of narrow categories of errors, rather than many different errors. If one were to evaluate the ontologies using a checklist of validity criteria or common errors, they would have gotten few black marks; yet a large proportion of their concepts was affected. If it can be shown that this pattern is typical, an ontology validation and correction strategy could be optimized accordingly.

Although a discipline of ontology validation and quality assurance is still evolving, our experiences so far have been positive and instructive. Potential future work includes the creation of an updated, comprehensive reference to ontology validity criteria, informed by a survey of previous case studies and the performance of additional new case studies.

Acknowledgement

This work was funded under NSF award #0934364 to the University of Maryland, Collaborative Research Establishing a Center for Hybrid Multicore Productivity Research..

References

1. Matuszek, C., Cabral, J., Witbrock, M., DeOliveira, J.: An Introduction to the Syntax and Content of Cyc. In Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, Stanford, CA, (2006).
2. OBO Download Matrix, http://www.berkeleybop.org/ontologies .
3. The OBO Flat File Format Specification, version 1.2, http://www.geneontology.org/GO.format.obo-1_2.shtml .
4. Cell Line Ontology version 1.0, May 2009, http://bioportal.bioontology.org/ontologies/39927/.
5. L. D. Cooper et al., "The Plant Ontology: Development of a Reference Ontology for all Plants," http://icbo.buffalo.edu/F2011/slides/AnatomyWorkshop/ FAOI2011.08.Cooper.pptx, 2011.
6. Mungall, C.: Cell Ontology 2012-12-13[STET – typo in year] release, http://sourceforge.net/mailarchive/message.php?msg_id=28544354 , December 15 2011.
7. OpenCyc, http://www.opencyc.org/, 2012.
8. Delete meaningless upper level terms, Cell-Ontology Issues Wiki, http://code.google.com/p/cell-ontology/issues/detail?id=1 .

Short Paper: Non-Taxonomic Concept Addition to Ontologies

Artemis Parvizi, Chris Huyck, and Roman Belavkin

Middlesex University
The Borough
London NW4 4RL

Abstract. Concept addition, an ontology evolution's edit operation, includes adding taxonomic (hierarchical structure) and non-taxonomic (concept properties) relations. Generating concept properties requires information extraction from various sources, such as WordNet. Other than semantic similarities generated by WordNet, self-information generated from existing non-taxonomic relations has aided non-taxonomic relation addition to ontologies. Evaluation is based on using an ontology as gold standard and detaching and reattaching the nodes. Non-taxonomic relation generation without accessing an enormous amount of information has proven to be quite difficult; the results displayed in this work are an indication of this difficulty.

Keywords: Ontology Evolution, Ontology Learning, Non-Taxonomic Relations, Concept Addition

1 Introduction

Ontology is commonly defined as a formal, explicit specification of a shared conceptualisation [14], and often has been used for modelling concepts of the world. Due to the experts' limitations of producing a complete image of the world with flexible boundaries for a domain, change is inevitable. Change in ontologies has some common causes [29]: change in the domain, change in the shared conceptualisation, or change in the specification. Ontology update has been a subject of debate for many years, and many methods have been proposed to address it. Ontology evolution and ontology learning are among these proposed methods. *Ontology evolution* is "the timely adaptation of an ontology to the arisen changes and the consistent propagation of these changes to dependent artefacts" [39], such as systems defined in [5, 30, 22, 40, 13, 4, 42, 19, 21]; *ontology learning* involves changing an ontology automatically or semi-automatically by consulting some structured data sources, such as databases; semi-structured data sources, such as WordNet, or Cyc; or some unstructured data sources, such as text documents and web pages [10]. A few examples of ontology learning systems can be found in [20, 9, 36, 27, 11, 41, 33].

Changing an ontology involves both changing the concepts and the relations. Ontology relations have been divided into two categories: taxonomic relations

such as `subClassOf` and `disjointWith` in OWL [2], and non-taxonomic relations which covers most of the other OWL relations. On one hand, taxonomic relations provide a structure to ontologies and are crucial. On the other hand, non-taxonomic relations by presenting meaning add depth to the ontology. Regardless of using the term ontology evolution or ontology learning, commonly, ontology update involves changing both taxonomic and non-taxonomic relations.

A fundamental design operation for having a successful ontology evolution application includes concept addition [24, 15]. To address concept addition, two approaches (Approach I (see Section 4.1) and Approach II (see Section 4.2)) have been introduced in which ontology graphs (see Section 2) and semantic similarity (see Section 3) have been employed.

2 Ontology Graph

The definition of an ontology in this paper is a set C of concepts and a set of relations $R_1, \ldots, R_n$, $R_i \subset C \times C$. Since multiple relations with different labels are allowed to exist in ontologies, labelled graphs also known as multigraphs ($G = (V, E_1, \ldots, E_n)$) with the set of vertices $V \iff C$ and a set of edges $E_i \iff R_i$ are a logical choice of representing them. A graph with the stated characteristics is called an *ontology graph* and is able to cover all important structural OWL ontology features including individuals, classes, relations, object properties, datatype properties, and restrictions [23]. The notion of ontology graph in this work is an extended version represented in [26, 16, 34, 17, 3, 25]; vertices represent concepts, individuals, restrictions, and values, and edges, include taxonomic OWL relations, such as `subClassOf` and `disjointWith`, and non-taxonomic relations.

3 Semantic Similarity

A successful ontology change application must be able to detect what needs to be changed, gather sufficient information about the element that needs to be changed, and finally decide how to implement change. Extracting relevant and sufficient information is crucial. In this work, WordNet [38] and Wikipedia as general purpose semi-structured data sources are consulted; they both are capable of generating semantic similarity distances between concepts. Semantic similarity between two or more concepts refers to the level of closeness that their meanings possess, and it is very difficult to acquire. It is common practice to use ontologies for computing the distance between two concepts and normalising the final result. In RiTa WordNet [18], the minimum distance between any two senses for the two words in the WordNet tree is returned and the result is normalised; if there is a similarity a number is retuned, and 1 if no similarity is found.

This work has generated semantic similarities from Wikipedia as well. Although many have mentioned that Wikipedia is much richer and a far better source [35, 7, 32, 37], the result acquired from Wikipedia were not as promising as WordNet. Often semantic Wikipedia APIs only consult the infoboxes for

generating semantic similarity; lack of word sense when extracting concepts is identified as another shortcoming [37].

4 Methodology

Ontology development is highly dependent on ontology experts, and domain experts. The perception of an expert about a correct or an incorrect relation may differ from another expert. This issue has contributed to the complexity of ontology development and update. Nonetheless, this work proposes that when adding a non-taxonomic relation, provided that the consistency of the ontology holds and the ontological statement is semantically correct, the new statement is as welcomed as any existing statement. For example when given the three concepts `Student`, `Library`, and `Group`, and the relation `memberOf`, an expert might generate `Student memberOf some Library`, `Student memberOf some Group`, or both. Absence of either of these two statements will not make the ontology incorrect but in certain circumstances it can be claimed that the ontology is less accurate. The same justification holds when a system is automatically generating non-taxonomic statements.

Commonly when generating non-taxonomic statements, a common approach is to provide a set of possible properties for each concept, rank them according to their frequencies, and finally according to some criteria select the highly probably one. However, this work does not intend to generate new properties for concept, but to assign an existing property to an input concept. Non-taxonomic relations can be classified into two general groups: object properties (intrinsic and extrinsic), and data-type properties [28]. The aim of this work is to generate intrinsic properties for a new input concept based on the existing intrinsic properties. The hypothesis is that siblings of a vertex in an ontology graph often have the same intrinsic properties assigned to different concepts.

In this work, the complete set of possible answers (Ans list) is generated, and the existing statements in the ontology (Cur list) are extracted. Ans list is a combination of an input concept I, the set of vertices $V = V_1, V_2, \ldots, V_n$, the set of edges $E = E_1, E_2, \ldots, E_n$, and the set of restrictions. Note that in this work only the two restrictions `some` and `only` are considered. Sample statements for the following approaches are as follows:

Existing Statement: $\texttt{V}_1\ \texttt{E}_1\ \texttt{some}\ \texttt{V}_2$
Generated Statement: $\texttt{I}\ \texttt{E}_1\ \texttt{some}\ \texttt{V}_3$

4.1 Approach I

The members of list Ans for an input concept I, m vertices, the two restrictions, and n edges will be $4 \times m \times n$ which comparative to list Cur are numerous. This approach consists of a number of filters to prune Ans list according to Cur list with the aid of various semantic similarities. To be able to apply semantic similarities, a random entropy or self-information approach has been selected.

Probability of the event of randomly connecting a to b by an R_i relation is defined by $P(e) = P((a,b) \in R_i)$. The prior probability therefore being $P(e) = \frac{1}{k}$, where k is the number of possible links $(a,b) \in R_i$. Given some semantic similarity distances (see Section 3) $s(a,b) \in [0,1]$, the posterior probability of a connection assuming a dependency between e and $s(a,b)$ is:

$$P(e \mid s(a,b)) \neq P(e)$$

Since $s(a,b)$ is a similarity distance (taking values in $[0,1]$ with 0 corresponding to the most similar), it can be assumed that the posterior probability of connection monotonically depends ($\propto$) on $1 - s(a,b)$:

$$P(e \mid s(a,b)) \propto 1 - s(a,b)$$

The monotonicity for two events $e_1 = (a,b)$ and $e_2 = (a,c)$ means the following:

$$\begin{aligned} s(a,b) \geq s(a,c) &\iff 1 - s(a,b) \leq 1 - s(a,c) \\ &\implies P(e_1 \mid s(a,b)) \leq P(e_2 \mid s(a,c)) \end{aligned}$$

The probability can be used to compute self-information as follows [6]:

$$\begin{aligned} h(a,b) &= -\log(P(e \mid s(a,b))) \\ &\approx -\log(1 - s(a,b))) \end{aligned} \tag{1}$$

The first filter is called hierarchy filtering; it is based on the patterns of the siblings of the input concept. A sibling is referred to a concept with a `disjoint-With` relation. This work focuses on non-taxonomic patterns. For the input concept I, assuming that one of the statements in Ans is IE_1onlyV_1, the patterns would be IE_1only and E_1onlyV_1. This approach only makes use of the forward patterns which in this example is E_1onlyV_1. Any member of the Ans list which does not contain the same pattern as one of the members of Cur list will be excluded from Ans. Also, if the input concept I and the first concept of a member of Cur list do not have the same parent, the statement will be excluded from Ans. Presuming both the pattern and the parent is matched, when the success rate of comparing the generated statement with all the members of Cur list is more than 50%, the statement will still remain in Ans, otherwise dropped. At this stage, only the statements with the patterns similar to the existing non-taxonomic statements remain.

From this point onwards, Equation 1 will aid the pruning process. For the second filter $Q_1 = h(I, E_1)$, $Q_2 = h(V_3, E_1)$, $Q_3 = h(V_2, E_1)$, and $Q_4 = h(V_1, E_1)$ are generated. The goal of this filter is to investigate $Q_1 + Q_2 \leq Q_3 + Q_4 \in [0.1]$; if in more than half the occurrences this function holds, then the generated statement will be accepted; otherwise rejected. The aim is for the self-information of the generated statement to be less than or equal to the self-information of the current statements.

For the third filter $Q_5 = h(I, V_1)$ and $Q_6 = h(V_2, V_3)$ are calculated. This filter will examine that in more than half the occurrences $Q_5 \leq Q_6 \in [0,1]$ holds.

The forth filter will generate $Q_7 = h(I, V_2)$ and $Q_8 = h(V_1, V_3)$; the relation $Q_7 \leq Q_8 \in [0, 1]$ must hold in more than half the occurrences for the generated statement to be accepted.

The last filter will generate the self-information among all the members of the generated and the current statement:

$$Q_i = h(Statement\ from\ Ans\ list, Statement\ from\ Cur\ list)$$

The result generated by Q_i are sorted and the k most similar statements selected. Tables 1 and 2 display the results when $k = 2$.

4.2 Approach II

The members of the *Ans* list have to be pruned according to the members of *Cur* list. A comparison between all the members of both lists is made. Providing that a statement from one of lists has the same relation and restriction (for example E_K `Some` or E_K `Only`) as the other list, the occurring pattern and its frequency is recorded. The list containing the patterns *Pat* will be sorted ascending with regard to the frequencies, and the top half selected. Those statements in *Ans* which do not contain these patterns will be omitted from the final answer pool. The statement V_1 E_1 `some` V_2 contains two patterns; (1) E_1 some V_2 and (2) V_1 E_1 some.

The aim of this step is to prune *Ans* list according to the patterns in *Cur* list; there is a trade off to this filter, some semantically correct statements will not be validated due to the low or lack of frequencies.

Hierarchy filtering as discussed in approach (I) will filter the remaining members of the *Ans* list. When the siblings of the input concept contain a non-taxonomic relation which have occurred in more than 50% of the cases and this taxonomic relation is among the remaining members of the *Ans* list, this statement will be accepted, otherwise rejected from *Ans* list.

4.3 Transitive Reduction

Both of the introduced approaches have the potential of producing transitive relations, which from the consistency point of view have to be eliminated. Inheritance through the hierarchy has to be modelled in an ontology graph. Transitive reduction on directed graphs is the answer to this problem. Presuming there is the possibility of representing information in the directed graph G with fewer arcs than the current amount, then that is the solution [1]. Graph G' will be the transitive reduction of G if it satisfies the following conditions:

1. A direct path between v and u in G' exists, if a direct path between v and u in G exists.
2. There is no graph with fewer arcs than G' satisfying the above condition.

For approach (II), since all the remaining members of the Ans list are selected, transitive reduction is applied after the last step. However, approach (I) is more complicated due to selecting the top k generated relations. Transitive reduction can be applied before or after the top k selection, which this work has adopted the latter. Regardless of the approach, in situations in which a child inherits a property and the algorithm identifies this transitive property, the property is dropped.

4.4 Evaluation

This work has adopted an evaluation mechanism based on precision and recall measurements [8, 12]. The strategy is to select a well-structured ontology and after converting it into an ontology graph, detach the vertices one by one; the system will attempt to reattach the vertex to the graph optimally with the original relations and at the original location [31]. A comparison between the number of *removed* edges in the original ontology graph (O) and the modified graph (O') is made. Assuming concepts c_1 and c_2 and relation R_k are present in O', the hypothesis is to examine O and determine whether c_1 and c_2 are related by R_k or not. Accepting the hypothesis indicates that O contains an edge corresponding to $c_1 R_k c_2$; rejecting is when there is no such edge in O. The overall count of correct edges in O' relative to the numbers of all edges in O' or O respectively will produce precision and recall. F-measure is a more just measurement since precision and recall are distributed evenly.

$$P(E', E) = \frac{|E \cap E'|}{|E'|} \qquad R(E', E) = \frac{|E \cap E'|}{|E|}$$

$$F(E', E) = 2 \times \frac{P(E', E)R(E', E)}{P(E', E) + R(E', E)}$$

Other than studying the effect of a single concept addition, the effect of adding a sequence of concepts also has to be studied. The order in which concepts a and b are added to the system has an effect on the non-taxonomic relations generated; generally, the semantic richness of the ontology is affected by the existing concepts and relations. This work has studied the effect of adding two ($p = 2$) and ten ($p = 10$) concepts to the ontology graph. Due to all the input concepts being known, the average of all the possible orders have been displayed.

Approaches (I) clearly has better results than approaches (II) excluding one exception. The more frequent a pattern, the higher the probability of it being selected; also, the closer the pattern in the hierarchy, the greater the likelihood of it being the final answer. The major difference between the two approaches other than the F-measure is in the number of statements being selected as the final answer. In the approach (I), the number of statements selected has a limit; as a result, fewer unmatched statements are selected. However, approach (II) has no limit on the number of generated statement, but at the same time more unmatched statements are in the final answer pool. The reason this paper is using the expression unmatched instead of incorrect is that studying the final

Table 1. The experimental results of non-taxonomic learning for approach (I). The results are displayed in percentage.

	p=1			p=2			p=10		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Pizza	0	0	unknown	0	0	unknown	0	0	unknown
Travel	25.0	50.0	33.33	25.0	50.0	33.33	25.0	50.0	33.33
Amino Acid	31.11	11.20	16.47	31.11	11.20	16.47	33.33	12.00	17.64
Career	20.00	26.66	22.85	20.00	26.66	22.85	15.00	20.00	17.14
Human and Pets	16.66	17.39	17.02	16.66	17.39	17.02	14.28	16.66	15.38
Movie	23.52	11.32	15.28	20.00	9.43	12.82	17.5	7.27	10.29
OBOE	0	0	unknown	0	0	unknown	0	0	unknown
University	19.56	14.51	16.66	19.56	14.51	16.66	11.36	8.06	9.43
Vehicle	14.28	18.18	16.0	14.28	18.18	16.0	23.07	27.27	25.0

Table 2. The experimental results of non-taxonomic learning for approach (II). The results are displayed in percentage.

	p=1			p=2			p=10		
	Precision	Recall	F-measure	Precision	Recall	F-measure	Precision	Recall	F-measure
Pizza	18.76	71.71	29.69	18.76	71.71	29.69	15.84	52.25	24.31
Travel	46.15	42.85	44.44	46.15	42.85	44.44	56.25	32.14	40.90
Amino Acid	52.50	63.00	57. 27	52.50	63.00	57. 27	59.42	41.0	48.52
Career	50	50	50	50	50	50	37.5	25.0	30.00
Human and Pets	52.77	39.58	45.23	52.77	39.58	45.23	57.57	39.58	46.91
Movie	45.16	70.0	54.90	48.83	70	57.53	49.33	61.66	54.81
OBOE	0	0	unknown	0	0	unknown	0	0	unknown
University	20.40	28.57	23.80	20.40	28.57	23.80	25.00	28. 57	26.66
Vehicle	0	0	unknown	0	0	unknown	0	0	unknown

results has shown that more than 50% of the unmatched statements are actually semantically and logically accurate, although, not present in the original answer pool. Nevertheless, Table (1) and 2 only display the result of correctly matched edges to the original graph.

5 Conclusion and Future Work

One ontology evolution operation is concept addition, which implies adding a concept by taxonomic and non-taxonomic relations. Commonly for changing an ontology some external information is required. In this work WordNet as an external source for generating similarities between concepts and relations has been

selected. The semantic similarities generated by WordNet, self-information produced from patterns within ontologies, and the hierarchical structure of ontologies are the basis of approaches introduced in this paper. The focus is on intrinsic properties; presuming that intrinsic properties already exist, the assumption is that an input concept is more likely to have the same intrinsic properties as its siblings. Evaluation is based on calculating the precision and recall of detaching a node from an ontology and attempting to reattach it. The results displayed in this paper are based on this evaluation technique. Due to the poor F-measures generated by the introduced approaches, an investigation into the cause of this poor performance revealed that more than 50% of the statements that were considered incorrect are actually semantically accurate. These results if generated by an ontology expert, could easily be regarded as correct.

The next step for this research is to generate more complex non-taxonomic relations, such as statements including conjunction and disjunction. Throughout the development of this work, the need for a ternary and a quaternary comparison has been visible. Such a comparison is essential for generating more meaningful ontology statements. Another future direction is to develop a source capable of ternary and quaternary comparison.

References

1. A V Aho, M R Garey, and J D Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.
2. H. Peter Alesso and Craig F. Smith. *Thinking on the Web: Berners-Lee, Godel and Turing*. Wiley-Interscience, New York, NY, USA, 2008.
3. Christoph Böhm, Philip Groth, and Ulf Leser. Graph-based ontology construction from heterogenous evidences. In *International Semantic Web Conference*, pages 81–96, 2009.
4. Silvana Castano, Irma Sofia Espinosa Peraldi, Alfio Ferrara, Vangelis Karkaletsis, Atila Kaya, Ralf Moeller, Stefano Montanelli, Georgios Petasis, and Michael Wessel. Multimedia interpretation for dynamic ontology evolution. *Journal of Logic and Computation*, 19(5):859–897, October 2009.
5. Fabio Ciravegna, Alexiei Dingli, David Guthrie, and Yorick Wilks. Integrating information to bootstrap information extraction from web sites. In *IJCAI'03 Workshop on Intelligent Information Integration*, pages 9–14, 2003.
6. Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.
7. Gerard de Melo and Gerhard Weikum. Menta: inducing multilingual taxonomies from wikipedia. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1099–1108, New York, NY, USA, 2010. ACM.
8. K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proceedings of the 5th International Semantic Web Conference (ISWC)*, 2006.
9. Takahira Yamaguchi Dept and Takahira Yamaguchi. Acquiring conceptual relationships from domain-specific texts. In *Proceedings of the Second Workshop on Ontology Learning OL'2001*, pages 0–2, 2001.

10. Lucas Drumond and Rosario Girardi. A survey of ontology learning procedures. In Frederico Luiz Gonçalves de Freitas, Heiner Stuckenschmidt, Helena Sofia Pinto, Andreia Malucelli, and Óscar Corcho, editors, *WONTO*, volume 427 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
11. E. Drymonas, K. Zervanou, and E. Petrakis. Unsupervised ontology acquisition from plain texts: the ontogain system. In *Proceedings of the 15th International Conference on Applications of Natural Language to Information Systems (NLDB)*, Wales, UK, 2010.
12. Jérôme Euzenat. Semantic precision and recall for ontology alignment evaluation. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 348–353, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.
13. Ademir Roberto Freddo and Cesar Augusto Tacla. Integrating social web with semantic web - ontology learning and ontology evolution from folksonomies. In *KEOD*, pages 247–253, 2009.
14. T Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
15. Mark Hall. Ontology integration and evolution. *SE Data and Knowledge Engineering*, 10, May 2004.
16. J. Hartmann, P. Spyns, A. Giboin, D. Maynard, R. Cuel, M. C. Suarez-Figueroa, and Y. Sure. D1.2.3 methods for ontology evaluation. Technical report, Knowledge Web Consortium, 2005. Version 1.3.1, Available at: http://knowledgeweb.semanticweb.org/, Downloaded 2005-05-10.
17. Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. *A Practical Guide To Building OWL Ontologies Using Prot eg e 4 and CO-ODE Tools*. The University Of Manchester, 1.2 edition, March 2009.
18. Daniel C. Howe. Rita: creativity support for computational literature. In *Proceeding of the seventh ACM conference on Creativity and cognition*, pages 205–210, New York, NY, USA, 2009. ACM.
19. Pieter De Leenheer. *On Community-based Ontology Evolution*. PhD thesis, Vrije Universiteit Brussel, Brussels, Belgium., 2009.
20. A. Maedche and S. Staab. Mining non-taxonomic conceptual relations from text. In *Proceedings of the 12th European Knowledge Acquisition Workshop (EKAW)*, Juan-les-Pins, France, 2000.
21. Yuxin Mao. A semantic-based genetic algorithm for sub-ontology evolution. *Information Technology Journal*, 9:609–620, 2010.
22. Diana Maynard, Diana Maynard, Wim Peters, and Marta" Sabou. Change management for metadata evolution. *International Workshop on Ontology Dynamics (IWOD) at European Semantic Web Conference*, 2007.
23. D.L. McGuinness and F. van Harmelen. *OWL web ontology language overview*. World Wide Web Consortium, Feb 2004.
24. Mohamed Mhiri and Faïez Gargouri. Using ontologies to resolve semantic conflicts in information systems design. In *Proceedings of The first International Conference on ICT and Accessibility*, Hammamet, Tunisia, April 2007. The first International Conference on ICT and Accessibility.
25. Victoria Nebot and Rafael Berlanga. Efficient retrieval of ontology fragments using an interval labeling scheme. *Information Sciences*, 179(24):4151 – 4173, 2009.
26. Chokri Ben Necib and Johann Christoph Freytag. Using ontologies for database query reformulation. In *ADBIS (Local Proceedings)*, 2004.
27. Mathias Niepert, Cameron Buckner, and Colin Allen. A dynamic ontology for a dynamic reference work. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '07, pages 288–297, New York, NY, USA, 2007. ACM.

28. Natalya Noy and Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, March 2001.
29. Natalya F. Noy and Mark A. Musen. Ontology versioning in an ontology management framework. *Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications]*, 19(4):6–13, 2004.
30. Philip O'Brien and Syed Sibte Raza Abidi. Modeling intelligent ontology evolution using biological evolutionary processes. In *IEEE International Conference on Engineering of Intelligent Systems*, pages 1–6, 2006.
31. Georgios Petasis, Vangelis Karkaletsis, Georgios Paliouras, Anastasia Krithara, and Elias Zavitsanos. *Ontology Population and Enrichment: State of the Art*, volume 6050 of *Lecture Notes in Computer Science*, pages 134–166. Springer Berlin Heidelberg, 2011.
32. Simone Paolo Ponzetto and Roberto Navigli. Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 2083–2088, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.
33. Janardhana Punuru and Jianhua Chen. Learning non-taxonomical semantic relations from domain texts. *Journal of Intelligent Information Systems*, pages 1–17, 2011. 10.1007/s10844-011-0149-4.
34. Sang Keun Rhee, Jihye Lee, and Myon-Woong Park. Ontology-based semantic relevance measure. In *Proceedings of the The First International Workshop on Semantic Web and Web 2.0 in Architectural, Product and Engineering Design*, 2007.
35. Navigli Roberto, Velardi Paola, and Faralli Stefano. A graph-based algorithm for inducing lexical taxonomies from scratch. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Spain, July 2011. IJCAI/AAAI.
36. David Sánchez and Antonio Moreno. Discovering non-taxonomic relations from the web. In *7th International Conference on Intelligent Data Engineering and Automated Learning. LNCS 4224*, pages 629–636, 2006.
37. Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 801–808, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
38. Michael M. Stark and Richard F. Riesenfeld. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*. MIT Press, 1998.
39. Ljiljana Stojanovic. *Methods and Tools for Ontology Evolution*. PhD thesis, University of Karlsruhe, Germany, 2004.
40. Carlo Torniai, Jelena Jovanovic, Scott Bateman, Dragan Gasevic, and Marek Hatala. Leveraging folksonomies for ontology evolution in e-learning environments. In *ICSC*, pages 206–213, 2008.
41. C. Trabelsi, A. Ben Jrad, and S. Ben Yahia. Bridging folksonomies and domain ontologies: Getting out non-taxonomic relations. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 369 –379, dec. 2010.
42. P. Wongthongtham, N. Kasisopha, and S. Komchaliaw. Community-oriented software engineering ontology evolution. *Internet Technology and Secured Transactions, 2009. ICITST 2009. International Conference for*, pages 1–4, November 2009.

Short Paper:

Deep Semantics in the Geosciences: semantic building blocks for a complete geoscience infrastructure

Brandon Whitehead,[1,2] Mark Gahegan[1]

[1]Centre for eResearch
[2] Institute of Earth Science and Engineering
The University of Auckland, Private Bag 92019, Auckland, New Zealand
{b.whitehead, m.gahegan}@auckland.ac.nz

Abstract. In the geosciences, the semantic models, or ontologies, available are typically narrowly focused structures fit for single purpose use. In this paper we discuss why this might be, with the conclusion that it is not sufficient to use semantics simply to provide categorical labels for instances—because of the interpretive and uncertain nature of geoscience, researchers need to understand how a conclusion has been reached in order to have any confidence in adopting it. Thus ontologies must address the epistemological questions of how (and possibly why) something is 'known'. We provide a longer justification for this argument, make a case for capturing and representing these *deep* semantics, provide examples in specific geoscience domains and briefly touch on a visualisation program called Alfred that we have developed to allow researchers to explore the different facets of ontology that can support them applying value judgements to the interpretation of geological entities.

Keywords: geoscience, deep semantics, ontology-based information retrieval

1 Introduction

From deep drilling programs and large-scale seismic surveys to satellite imagery and field excursions, geoscience observations have traditionally been expensive to capture. As such, many disciplines related to the geosciences have relied heavily on inferential methods, probability, and—most importantly—individual experience to help construct a continuous (or, more complete) description of what lies between two data values [1]. In recent years the technology behind environmental sensors and other data collection methods and systems have enabled a boom of sorts in the collection of raw, discrete and continuous geoscience data. As a consequence, the operational paradigm of many conventional geoscience domains, once considered data poor, now have more data than can be used efficiently, or even effectively. For example, according to Crompton [2], Chevron Energy Technology Corporation had over 6000 Terabytes of data, and derived products such as reports, and is rapidly expanding. This data deluge [3], while significant in its affect on capturing information related to complex earth science processes, has become a Pyrrhic victory for geoscientists from a computational perspective.

The digital or electronic facilitation of science, also known as eScience [4] or eResearch, coupled with the science of data [5] is fast becoming an indispensable aspect of the process of Earth science [6–8]. There are exemplar projects such as OneGelogy[1], which translates (interoperates) regional geologic maps in an effort to create a single map of the world at 1:1 million scale; as well as the Geosciences Network[2] (GEON) which houses a vast array of datasets, workflows, and tools for shared or online data manipulation and characterisation. Further, the National Science Foundation (in the U.S.A.) has funded EarthCube[3] which seeks to meld the perspectives of geoscientists and cyberscientists to create a framework for locating and interoperating disparate, heterogeneous information about the entire Earth as a comprehensive system. The major contributions that eScience can make is by providing ways to communicate the semantics, context, capabilities and provenance of the datasets, workflows, information and tools in order for researchers to have a firm understanding of the artefacts they are using, and how they are using them.

In this paper, we illustrate how multiple, multi-faceted semantic models are coordinated under the linked data paradigm to better reflect how geoscience researchers situate concepts with their own knowledge structures in an effort to contextualise observations, phenomena and processes. We look to expose which semantic, or ontological, commitments are needed to glean how science artefacts relate to researchers, methods and products (as data, or via theory) in order to transfer what is known about a place, and how it is known, as a useful analog for geoscience discovery. We use an interactive computational environment, known as *Alfred*, to view disparate ontologies that carry pieces of this 'knowledge soup' [9] as facets, and expose the relationships for discovery of new knowledge.

2 Geoscience Background

The geosciences are far from exact; the earth as a living laboratory provides plenty of challenges, not least to the task of representing and communicating semantics. While geoscientists are remarkable in their ability to utilise disparate knowledge in mathematics, physics, chemistry and biology to create meaning from observed phenomena, their theories are bound by the inherent problems associated with scale and place, cause and process, and system response [10]. The Earth's phenomena are complex, they often exhibit statistically unique signatures with several stable states while mechanical, chemical and biologic processes work in tandem, or asynchronously. Due to these often contradictory complications it has also been suggested that the Earth sciences exemplify a "case study to understand the nature and limits of human reasoning, scientific or otherwise" [11]. Adding to the complexity, "Geologists reason via all manner of maps, outcrop interpretation, stratigraphic

[1] http://www.onegeology.org/
[2] http://www.geongrid.org
[3] http://earthcube.ning.com/

relationships, and hypothetical inferences as to causation" [12] and they do this simultaneously across geographic and temporal scales.

In order to discern the categories and components of the Earth as a system, the geoscientist requires a trained eye, what anthropologists call "professional vision" [13], which often necessitates years of experience and mentoring. This contextualised view of the world uses a long view of time, and becomes adept at distinguishing infrequent catastrophic events from those more frequent via the feedback loops between processes and components [13]. However, these feedback loops are often not well understood due to the fragmented nature of geoscience observation and data. This has required the geoscience community of practice to develop the means by which their observations are understood. Most notably, instead of constructing a specific research question and testing it, geoscientists often use the method of 'multiple working hypotheses' [14] and work toward reducing what is not known, instead of working towards some axiomatic truth. Indeed, the ability to abstract earth processes to a rational metaphoric justification could be considered an art form.

As such, geology is often referred to as an interpretive science [15]; where empirical evidence is not possible, a story often emerges. Interpreting meaning in the geosciences revolves heavily around the inherent allusion in hypothesis, methods, models, motivations, and often more importantly, experience. Understanding the knowledge any researcher creates requires understanding that person's research methods and the rationale behind their decision processes, which requires the ability for knowledge components to change roles as one tries to demystify the scale in context and perceptions from which they are constrained. Often, what is determined to be a result is steeped in probability as a function of a desired resource. To date, the research and research tools used throughout geoscience domains are largely situational; capturing tightly coupled observations and computations which become disjointed when the view, filter, or purpose is altered, even slightly, to that which is more representative of an earth system science.

3 Semantic Modelling in the Geosciences

As the previous section suggests, the semantic nature of geoscientific ideas, concepts, models, and knowledge is steeped in experiential subjectivity and often characterised by what can or cannot be directly observed, directly or indirectly inferred, and, in many cases, the goals of the research. As the Semantic Web [16] has gained traction and support, a subset of Earth science researchers have been intrigued by the possibility of standards, formal structure, and, ultimately, ontologies in geoscience domains, mainly because, as Sinha et al., have stated, "From a scientific perspective, making knowledge explicit and computable should sharpen scientific arguments and reveal gaps and weaknesses in logic, as well as to serve as a computable reflection of the state of current shared understanding" [17].

As evidenced by the dearth of semantic models, or ontologies, in the earth sciences [18], the often-conflicting ideals and knowledge schemas are proving to be significant hurdles for ontological engineers. Most of the semantic models in Earth science communities would be considered weak [19], lightweight (sometimes referred to as 'informal') [20, 21] or implicit [22]. These include taxonomies, or controlled vocabularies—like the American Geophysical Union's (AGU) index of terms,[4] glossaries [23], thesauri [24], or a typical data base schema. Conversely, semantic models created with the aspiration of eventuating to strong, heavyweight or formal ontologies are limited. In cases where published formal domain ontologies do exist [25], they are often not openly available within the community.

One openly available ontology of note is the upper-level ontology SWEET: Semantic Web for Earth and Environmental Terminology [26]. This formal ontology was created to tag the huge repositories of satellite imagery created and housed by NASA. As a result, the concepts used in SWEET are very high level and the granularity of the ontology is, in most cases, not detailed enough to differentiate between the thousands of resources an active research geoscientist might find useful,

There is a middle ground between the two ends of the semantic spectrum, in the form of Mark-up Languages, which is quite promising. In the geosciences, two exemplar Mark-up Languages do exist; the Geography Mark-up Language (GML) [27], and Geoscience Mark-up Language (GeoSciML) [28]. The two languages were created to serve mainly as a translation schema for data sharing and interoperability, and do provide a level of formalisation and weakly typed relationship structure.

4 A Case for *Deep* Semantics

In this research we use the relative lack of formalised structures in the geosciences as an opportunity to start from scratch and take a slightly different approach to ontological engineering in the domain. We try get away from the highly restrictive, monolithic, overarching structures and focus on a more complete picture of the relational patterns of geoscientific artefacts. To summarise Gahegan et al., [29]: we are looking to expose the 'web of multi-faceted interactions' between observations, theory, data, motivations, methods, tools, places and people. To focus the modelling effort, we asked the following questions: How is something known? Which entities support a research artefact? Who has been publishing about a topic, concept, place, research method or data product? What was the inference path a geoscientist traveled from a piece of evidence to an interpretation?

As these questions might suggest, a *deep* semantic support structure should provide a conceptual richness that permeates the depth of a specialised set of concepts and provide a mechanism for defining how an artefact came to be represented. A *deep* semantic structure should provide enough specificity in the concepts and relations that

[4] http://www.agu.org/pubs/authors/manuscript_tools/journals/index_terms/

the terms can be used to differentiate complex but real situations via the written materials describing these situations, not simply how it was labelled, or tagged, by the creator, librarian or data curator. Exposing this story behind the data, or, more formally, the epistemological connections, *deep* semantics works towards constraining the conceptual uncertainty of the procedural knowledge by explicitly representing and exposing the semantics of research artefacts as the scientist has orientated them for his/her evidence, and thus, the resultant interpretation. This effectively frees the research scientist to focus on the declarative knowledge supporting the probabilities in the numerical components.

The ability to locate resources has become increasingly important as data storage continues to increase. What carries a heavier weight is the ability to locate a data product at the time it is most useful, by being able to distinguish a resource's *when* and *where* relatively rapidly. With a *deep* semantic view, we are able to begin pursuing the *why* and *how* of the conceptual structures that support geoscientific knowledge and discovery. In the information sciences this is often referred to as precision and recall. Deep semantics adds epistemological underpinnings and a level of context to precision and recall while adding facets to the constraint and delineation mechanisms.

5 Ontology Inception and Use

Building the relationship structures, as described in this section, of the disparate parts of geoscience research artefacts creates a contextualised, and in this case visual, representation of the network of ontological components that support a concept. We treat every ontological component as linked data supported by domain specific terminological ontologies [30]. We use the full version of the Web Ontology Language (OWL Full) to promote emergent constraints via relationships when possible. OWL Full was chosen due to its compatibility with other modeling approaches, most notably Resource Description Framework (RDF), as well as reducing the restrictions on class definitions. The latter is necessary in the Earth sciences as it is quite common to find a concept, or identifier, that is a class name as well as an instance. In addition, given the nature of geoscience knowledge, it should not be logically impossible to arrive at a conclusion that is not yet known to the system through the ontological framework. We felt this fits more in line with the process of Earth science, which relies quite heavily on reducing what is not known rather than enforced, top-down, logical constraints depicting what is known axiomatically. It is this connected interworking of heterogeneous semantic models ranging from weak to strong, lightweight to heavyweight, and informal to formal which join together as linked data, that we have come to refer to as *deep* semantics. The remainder of this section describes how each individual ontology was constructed.

5.1 Basin and Reservoir ontology

We endeavoured to create a framework for formal geoscience knowledge as it applies to sedimentary basins and reservoirs in the energy and petroleum industry under the aegis of recognised industry Subject Matter Experts (SMEs). The SMEs participated in knowledge acquisition exercises [31] orchestrated to discuss fundamental concepts and their meanings as interpreted and explained through their formal and experiential mastery. As concepts emerged, they were explicitly described, often through diagrams and examples, to the satisfaction of the other participants. Prior to each workshop, a set of concepts had been extracted from a survey of applicable literature in the domain to serve as exemplars for the types of concepts found in research artefacts that differentiate and describe specific geoscience situations and models. These concepts were periodically re-introduced to the SMEs to ensure structure that was being created had semantically tenable end-points. This process allowed interrelationships between fundamental and domain-level concepts to be exposed and characterised. As the exercise progressed, clusters of concept and relationship types became apparent. The open nature of the knowledge acquisition exercise allowed the participants to navigate through the conceptual neighbourhood that they had created. As such, there are areas of the concept space that are defined more rigorously than others.

The workshops culminated two ontological frameworks: a Basin ontology [32, 33] and a Reservoir ontology. The Basin ontology focuses on concepts corresponding to basin characterisation. The core concepts are related to properties and other classes through select earth processes (e.g., the Basin class is related to the Strata class via a tectonic processes, such as subsidence). The Reservoir ontology was created quite similarly as the Basin ontology, with the exception being that the contributing SMEs were well versed in petroleum reservoir characterisation and modeling instead of basin characterisation.

The Basin and Reservoir ontologies have been created to interoperate with each other to coordinate the delineation of scale dependent ambiguities in research artefacts. To further promote semantic interoperability, both of these ontologies have natural contact points for semantic correlation with upper-level earth science ontologies in the public-domain, such as SWEET, as well as with other domain-specific ontologies from hydrocarbon exploration and production, to hydrologic and paleoclimate modeling, should they become available.

5.2 Agent and Resource ontology

Two of the more important facets of this research are the actual research artefacts and the researchers, or creators, of those artefacts. Fortunately, librarians have already spent a significant amount of time developing a standard for metadata that captures the types of information that we wanted to capture from resources. We used the Agent profile from the Dublin Core Metadata Initiative[5] (DCMI) to describe authors, contributors, software, companies, and research groups. There are other types of agents, of course, and DCMI is set up to handle these distinctions, but for our

[5] http://dublincore.org/

purposes a subset was all that was required. The Resource profile from DCMI is used to describe any artefact produced by research. This can include publications, abstracts, presentations, and data products. Again, the schema for the DCMI framework allows for a plethora of types, but a subset was all that was required here.

5.3 Task ontology

The Task ontology was constructed to provide a framework for actions that are completed during research. These include observations, methods and processes like data collection, data manipulation, statistical methods, etc. Items in the Task ontology link to Resources as outputs and inputs, and to Agents as creators, contributors, reviewers, etc. The concepts in this specification are often chained together to create large structures, and are helpful in delimiting clusters of information. The Task ontology was created by, first, describing a small set of exemplar concepts that related strongly to key components of the semantic models described in the previous two sections. Once the initial concepts were introduced, the structure was extended by defining known superclasses and subclasses, and then supplanting those core concepts with text mining utilising basic natural language processing principles.

5.4 Oilfield Glossary and World Oil and Gas Atlas ontology

The Schlumberger Oilfield Glossary[6] is a fantastic on-line resource covering an expansive number of topics. The Oilfield Glossary ontology was constructed from harvesting the information hosted on this web site. Due to research limitations, it was more beneficial to create a local copy of this data and convert that to a series of triples than to develop a script to query the site interactively. During the data conversion, all partial relationships within the structure, and the links to the corresponding web page were preserved.

The World Oil and Gas ontology was created by manually entering information provided in the summaries, graphs, and tabular data, as depicted in the World Oil and Gas Atlas [34], into an electronic format. Once in an electronic format, a script was generated to alter the format, along with a little manual editing, to OWL.

6 Early Results: Powder River Basin Use Case

This use case illustrates how research artefacts associated with the Powder River Basin, located in the central part of the U.S.A., can be visualised and navigated via a knowledge computation platform referred to here as *Alfred*. This platform allows for navigating and manipulating disparate multifaceted structures in one graph space. The user loads an ontology into the system, which is then represented as a facet. Any facet can be docked to any length of the graph border. Through docking a facet (up to four), *Alfred* provides a space for the user to follow their interests in their linked data exploration.

---

[6] http://www.glossary.oilfield.slb.com/

We proceed from the perspective of a research scientist with an interest in the Powder River Basin. The user enters "Powder River" into *Alfred's* the search field. The resulting graph, shown in Figure 1, shows two concepts matching the search term within a neighbourhood of related concepts. One of the Powder River concepts (highlighted with a yellow outer ring) is symbolised using a gold circle coupled with a black 'basin' object from a scalable vector graphic (SVG). The edge pointing to the yellow circle labeled Basin, denotes it is an instance of the *Basin* class (yellow disc) from the Basin ontology. The other symbol labeled Powder River is a grey triangle which signifies it as a member of the World Oil and Gas ontology (in the structure, these two concepts are in fact connected via an *owl:sameAs* relation, but this type of relation has been suppressed in the current view for readability). Three conference abstracts, symbolised by a red circle, with an SVG in the shape of a book, relate via a *references* edge to the Powder River concepts, as well as a few concepts symbolised by black diamonds, which are delineating concepts from the Oilfield Glossary ontology.

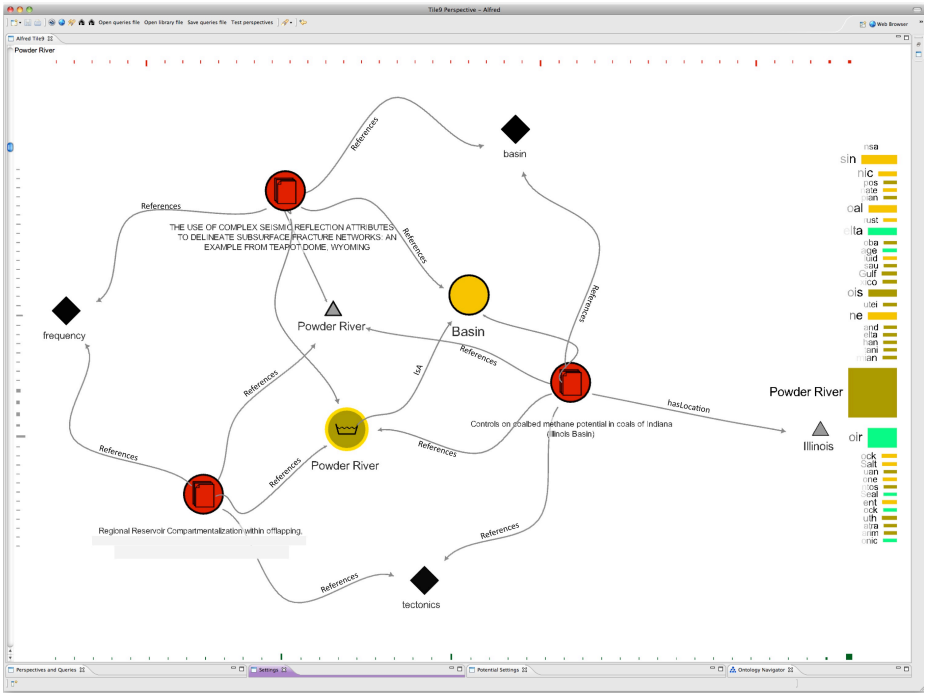

Fig. 1. Local graph neighbourhood of the concept representing the Powder River Basin. The current view shows three published artefacts, how the concept Powder River is linked in the hierarchy (it is an instance of Basin) as well as a few terms from the Oilfield Glossary.

At this stage, the user has a few options. The user can select something from the graph, adjust the filter settings to increase the type and/or level of information displayed in the graph, or start a new search. To continue with the example in the use case, we assume the users interest has been piqued by one of the research artefacts

sharing a relationship with the Powder River Basin concept. If the user were interested in seismic reflection data, they might select the artefact purporting to deal with complex seismic reflection attributes (red disc with book SVG, lower left) via a double click.

Upon this click action, the graph re-centres itself using the user selected node as the central concept, as depicted in Figure 2. This selection reveals a deeper structure associated with that particular artefact. In this view, the creator of the artefact, symbolised by a dark green circle surrounding a SVG of a person, has emerged along with several concepts found in the Oilfield Glossary. The relationship to the Powder River and Basin concepts have persisted to the new concept layout (lower middle of Fig. 2). Several concepts from the Task ontology (blue circles) have emerged, potentially signifying relationships to the data (seismic data) as well as concepts related to analysis mechanisms (phase coherence) associated with this particular research artefact. This view also provides the user with contextually similar research artefacts by displaying the research outputs that share a relation with other concepts in the known ontologies.

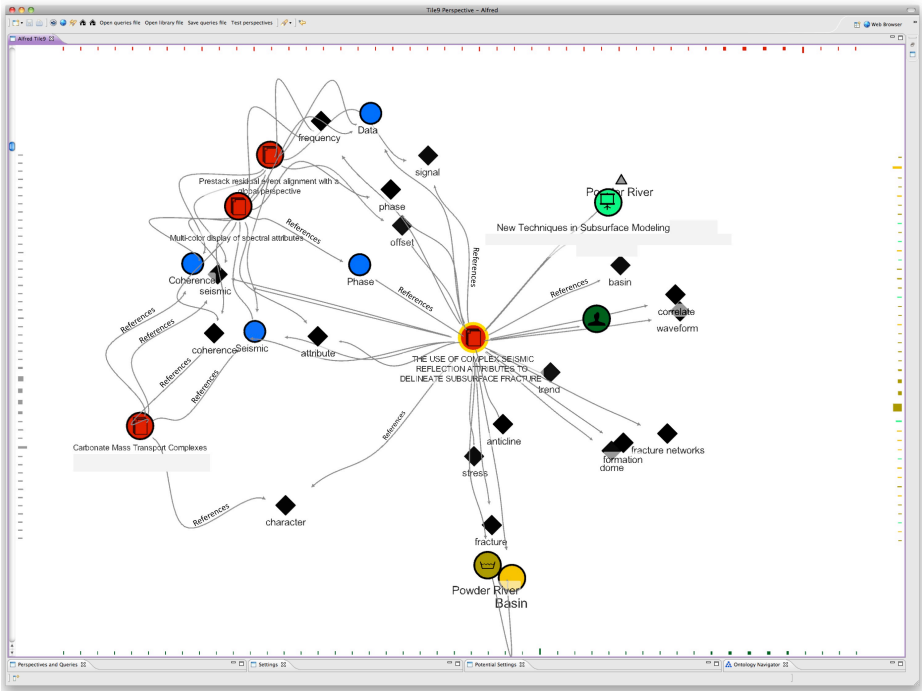

Fig. 2. Local graph neighborhood of a research artefact related to the Powder River Basin. The current view shows the artefacts creator, as well as other concepts from the semantic structures known to the system.

In the example depicted in Figure 2, three research artefacts appear to share several relationships (mostly a *reference* edge) with concepts found in the Oilfield Glossary, as well as the Tasks ontology. This clustering suggests there are other research

products that have utilised the same, or similar, methods and data that were used in the research artefact of interest. This is worth mentioning here as the related data, tasks, and concepts allow the user to explore and glean the concepts and structures that support a research artefact. The ability to navigate, what has become, the epistemological lineage of a research artefact cultivates a formal representation of the symbiotic components of geoscience research products and geoscience knowledge.

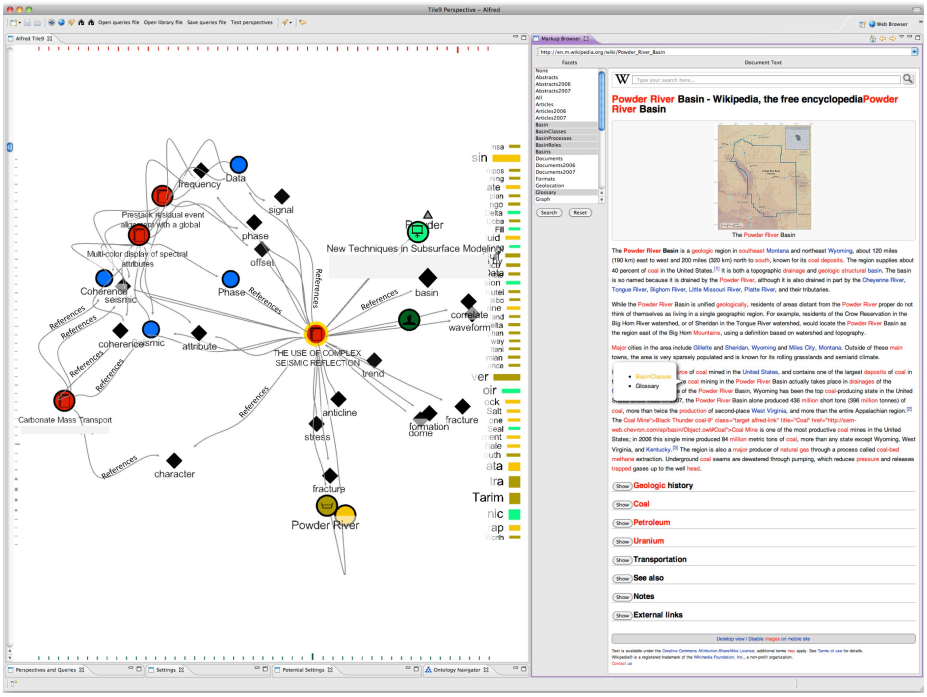

Fig. 3. A local graph neighbourhood shown with a web page that is marked up with red text using the concepts from the ontologies loaded into the system. The user can go back and forth between graph space and the web page in order to better refine the context and scope of research artefacts and web enabled content.

At this final stage of the use case, we illustrate how the conceptual neighbourhood of the graph can synchronise with other web enable components, in this case browser content. As portrayed in Figure 3, a user can open a web page and search for known semantic components on that page. When the search has completed, all known concepts are now displayed with red text within the browser. Further, by hovering over any syntactic component on the corresponding web page (in this instance, it is the mobile version of the Wikipedia page for the Powder River Basin) the user is presented with a pop-up dialogue populated by the referring ontology. If a term is situated in multiple ontologies, each one will be listed in the pop up, with the text providing a live link back to the graph space. In this way, a user could bring their conceptual neighbourhood with them as they peruse web content and use the highlighted text references to help filter for relevance. This has proved to be

particularly helpful with the increasing number of peer reviewed publications available in a web friendly format.

The Powder River Basin use case illustrates how *deep* semantics can benefit geoscientists by providing a mechanism to visualise and explore the components that comprise a knowledge construct. When a geologist purports to know how to characterise a particular basin, other geologists and engineers naturally want to know what data and analysis methods were used to support that interpretation. How was the stratigraphy interpreted? What was the timing of the tectonic events? What is the burial history? *Deep* semantics allows other geoscientists to explore these supporting entities and the decisions that were made along the path to any particular explication.

7 Concluding Remarks

Geoscience ontologies are typically quite lightweight, or implicit, and are engineered for one specific purpose. As such, the semantic structures in the geosciences fail to capture the complexities and intricacies inherent in the domain knowledge. Ontologies like SWEET are a great start at a general upper level structure for geoscience domains, but other than providing a label for instances, these structures are far removed from capturing the level of detail necessary to empower domain scientists, or knowledge engineers, with useful components for day-to-day meaningful research activities. In this paper we illustrate how a *deep* semantic structure serves to differentiate research products by capturing epistemological commitments of geoscience research artefacts using ontologies throughout the spectrum of formalisation. This deep semantic structure provides the conceptual backbone for geoscientific search, discovery and enquiry.

Acknowledgments. The authors would like to thank Dr. Will Smart and Sina Masoud-Ansari, at the University of Auckland's Centre for eResearch, for their contributions to the Alfred framework used to illustrate the use case in this paper. The authors would also like to acknowledge the generous support of the New Zealand International Doctoral Research Scholarship (NZIDRS), which helped make this research possible.

8 References

1. Brodaric, B., Gahegan, M.: Experiments to Examine the Situated Nature of Geoscientific Concepts. Spatial Cognition and Computation. 7, 61–95 (2007).
2. Crompton, J.: Putting the FOCUS on Data. W3C Workshop on Semantic Web in Oil & Gas Industry. , Houston, USA (2008).
3. Bell, G., Hey, T., Szalay, A.: Beyond the Data Deluge. Science. 323, 1297–1298 (2009).
4. Hey, T., Trefethen, A.E.: Cyberinfrastructure for e-Science. Science. 308, 817–821 (2005).
5. Hey, T., Tansley, S., Tolle, K. eds: The Fourth Paradigm: Data-intensive scientific discovery. Microsoft Research, Redmond, Washington (2009).
6. Sinha, A.K. ed: Geoinformatics: Data to Knowledge. Geological Society of America (2006).

7. Sinha, A.K., Arctur, D., Jackson, I., Gundersen, L. eds: Societal Challenges and Geoinformatics. Geological Society of America, Boulder, CO (2011).
8. Keller, G.R., Baru, C. eds: Geoinformatics: Cyberinfrastructure for the Solid Earth Sciences. Cambridge University Press, Cambridge, UK (2011).
9. Sowa, J.F.: Crystallizing Theories out of Knowledge Soup. In: Ras, Z.W. and Zemankova, M. (eds.) Intelligent Systems: State of the Art and Future Directions. pp. 456–487. Ellis Horwood, New York (1990).
10. Schumm, S.A.: To Interpret the Earth: ten ways to be wrong. Cambridge University Press, Cambridge, UK (1991).
11. Raab, T., Frodeman, R.: What is it like to be a geologist? A phenomenology of geology and its epistemological implications. Philosophy and Geography. 5, 69–81 (2002).
12. Baker, V.R.: Geosemiosis. GSA Bulletin. 111, 633–645 (1999).
13. Kastens, K., Manduca, C.A., Cervato, C., Frodeman, R., Goodwin, C., Liben, L.S., Mogk, D.W., Spangler, T.C., Stillings, N.A., Titus, S.: How Geoscientists Think and Learn. Eos. Trans. AGU. 90, 265–266 (2009).
14. Chamberlin, T.C.: The Method of Multiple Working Hypotheses. Science. 148, 754–759 (1899).
15. Frodeman, R.: Geological reasoning: Geology as an interpretive and historical science. GSA Bulletin. 107, 960–968 (1995).
16. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American. 284, 34–43 (2001).
17. Sinha, A.K., Ludäscher, B., Brodaric, B., Baru, C., Seber, D., Snoke, A., Barnes, C.: GEON: Developing the Cyberinfrastructure for the Earth Sciences. (2003).
18. Babaie, H.A.: Ontological relations and spatial reasoning in earth science ontologies. In: Sinha, A.K., Arctur, D., Jackson, I., and Gundersen, L. (eds.) Societal Challenges and Geoinformatics. pp. 13–27. Geological Society of America, Boulder, CO (2011).
19. Obrst, L.: Ontologies for semantically interoperable systems. Proceedings of the twelfth international conference on Information and knowledge management. pp. 366–369. ACM, New Orleans, LA (2003).
20. Sowa, J.F.: Future Directions for Semantic Systems. In: Tolk, A. and Jain, L.C. (eds.) Intelligence-Based Systems Engineering. pp. 23–47. Springer-Verlag Berlin (2011).
21. Giunchiglia, F., Zaihrayeu, I.: Lightweight Ontologies. In: Liu, L. and Özsu, M.T. (eds.) Encyclopedia of Database Systems. pp. 1613–1619. Springer, US (2009).
22. Sheth, A., Ramakrishnan, C., Thomas, C.: Semantics for the Semantic Web: The Implicit, the Formal and the Powerful. International Journal on Semantic Web & Information Systems. 1, 1–18 (2005).
23. Neuendorf, K.K.E., Mehl, Jr., J.P., Jackson, J.A.: Glossary of Geology. American Geosciences Institute (2011).
24. Deliiska, B.: Thesaurus and domain ontology of geoinformatics. Transactions in GIS. 11, 637–651 (2007).
25. Zhong, J., Aydina, A., McGuinness, D.L.: Ontology of fractures. Journal of Structural Geology. 31, 251–259 (2009).
26. Raskin, R.G., Pan, M.J.: Knowledge Representation in the Semantic Web for Earth and Environmental Terminology (SWEET). Computers & Geosciences. 31, 1119–1125 (2005).
27. Lake, R., Burggraf, D.S., Trninić, M., Rae, L.: Geography Mark-Up Language (GML). John Wiley & Sons, Ltd, West Sussex, England (2004).
28. Sen, M., Duffy, T.: GeoSciML: Development of a generic Geoscience Markup Language. Computers & Geosciences. 31, 1095–1103 (2005).
29. Gahegan, M., Luo, J., Weaver, S.D., Pike, W., Banchuen, T.: Connecting GEON: Making sense of the myriad resources, researchers and concepts that comprise a geoscience cyberinfrastructure. Computers & Geosciences. 35, 836–854 (2009).
30. Sowa, J.F.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Course Technology Cengage Learning, Boston, MA (2000).
31. Ribes, D., Bowker, G.: Between meaning and machine: Learning to represent the knowledge of communities. Information and Organization. 19, 199–217 (2009).
32. Whitehead, B., Gahegan, M., Everett, M., Hills, S., Brodaric, B.: Fostering the exchange of geoscience resources for knowledge exploration and discovery. Proceedings of eResearch Australasia Conference. p. 2p. , Gold Coast, Australia (2010).
33. Everett, M., Hills, S., Gahegan, M., Whitehead, B., Brodaric, B.: Improving Semantic Interoperability through the Development of a Reusable E&P Earth Science Ontology. American Association of Petroleum Geologists (AAPG) Annual Convention and Exhibition. , Houston, USA (2011).
34. Guoyo, L.: World Atlas of Oil and Gas Basins. WIley-Blackwell, Hoboken, NJ, USA (2010).

Short Paper: Assessing Procedural Knowledge in Open-ended Questions through Semantic Web Ontologies

Eric Snow, Chadia Moghrabi and Philippe Fournier-Viger

Département d'informatique, Université de Moncton, Moncton, Canada
{eric.snow,chadia.moghrabi,philippe.fournier-viger}@umoncton.ca

Abstract. This paper presents a novel approach for automatically grading students' answers to open-ended questions. It is inspired by the OeLE method, which uses ontologies and Semantic Web technologies to represent course material. The main difference in our approach is that we add a new category of concepts, named *functional concepts*, which allow specifying an ordering relation between concepts. This modification allows assessing procedural knowledge in students' answers by grading the ordering of these concepts. We present an example for grading answers in a course about computer algorithms, and report the corresponding results.

Keywords: E-Learning, Computer-Assisted Assessment (CAA), Ontology, Semantic Web, Procedural Knowledge.

1 Introduction

Assessing the students' learning in an e-learning environment often relies on multiple choice or fill-in-the-blank questions, which only trigger the lowest level (*Knowledge*) of Bloom's taxonomy [1] of knowledge acquisition. As we shall see in Section 2, several attempts have been made to incorporate open-ended questions in online assessment, which would possibly trigger the higher levels of Bloom's taxonomy (*Synthesis* and *Evaluation*) in the students' learning.

However, grading open-ended questions by hand can be time-consuming. To build an e-learning environment that can automatically grade free-text answers, a variety of techniques have been used, such as Information Extraction (IE) [4-5], Natural Language Processing (NLP) [6-11], or statistical techniques [13-15].

Our approach resembles that of the OeLE system [2]. This system also uses NLP to assess the level of understanding of the students. Course material is represented in an ontology and encoded in the Web Ontology Language (OWL). The use of Semantic Web technologies allows the sharing and reusing of course ontologies, thus potentially reducing the time spent designing the ontologies. This allows for a deeper understanding of the text than more superficial statistical techniques. Automatic assessment is much faster, and hopefully done more objectively, than manual scoring. The OeLE system has been used in two online courses, *Design and Evaluation of Didactic Media, and Multimedia Systems and Graphical Interaction.*

OeLE successfully assesses the semantic content of the students' answers if the answers contain static expressions of facts about didactic media or multimedia systems. However, when applying it to the assessment of a computer algorithms course, we observed that the ordering of the elements in students' answers is not taken into account. It is crucial that this ordering be considered because to describe how an algorithm works, certain concepts should be stated in a specific order. In this paper, we address this challenge by proposing a new approach in which we introduce the idea of *functional concepts*. The course ontology then incorporates ordering information about a subset of these functional concepts. The assessment process is modified to take into account the ordering of these concepts in the students' answers and adjust their grade accordingly. The novelty of our work is in applying a hybrid approach combining the OeLE system with functional concepts to assess students' answers in domains using highly procedural knowledge.

Section 2 of this paper is a review of other automatic free-text assessment systems. We only focus here on short-answer assessment systems where reference texts are tailored to the course material, although some other systems have also been developed for essay scoring, where more general texts about a topic are used for training. Section 3 presents the general methodology, followed by our preliminary results in Section 4. We conclude the paper in Section 5 with some future work which we are investigating.

2 Related Work

This section presents previous and ongoing research in automatic short-answer assessment. A good review of many of these systems can be found in [3]. Although these systems do not take advantage of Semantic Web ontologies, they contain nonetheless functionalities and techniques useful to our system.

Some systems compare students' answers to the ideal answer supplied by the teacher. For instance, Automated Text Marker [4] uses a pattern-matching technique. It has been tested in courses on Prolog programming, psychology and biology. Automark [5] uses IE techniques to grade students' answers by comparing them to a mark scheme template provided by the teacher. It achieved 94.7% agreement with human grading for six of the seven science-related questions asked on a test exam.

Some systems require teachers to provide training sets of marked student answers. For example, Auto-marking [6] uses NLP and pattern-matching techniques to compare students' answers to a training set of marked student answers. This system obtained 88% of exact agreement with human grading in a biology course. Bayesian Essay Test Scoring System (BETSY) [7] uses naive Bayes classifiers to search for specific features in students' answers. In a biology course, it achieved up to 80% accuracy. CarmelTC [8] uses syntactic analysis and naive Bayes classifiers to analyze essay answers. On an experiment with 126 physics essays, it obtained 90% precision and 80% recall. The Paperless-School Marking Engine (PS-ME) [9] is commercially available and requires a training set of marked answers. The system uses NLP to grade the students' answers in addition to implementing Bloom's taxonomy heuris-

tics. However, the exact implementation is not disclosed. C-rater [10] uses a set of marked training essays to determine the students' answers grade using NLP. In a large-scale experiment of 170,000 answers to reading comprehension and algebra questions, it achieved 85% accuracy. In [11], a combination of NLP and Support Vector Machines is used to classify answers into two classes (above/below 6 points out of 10). It obtains an average of 65% precision rate (the only reported metric).

The MultiNet Working Bench system [12] uses a graphical tool to represent the students' knowledge visually. It compares the semantic network extracted from the student answer to that submitted by the teacher. Verified parts of the network are displayed in green, while wrong or unverified parts (not supported by logic inference) are displayed in red.

Other systems rely on Latent Semantic Analysis (LSA). For example, Research Methods Tutor [13] uses LSA to compare the students' answers to a set of expected answers. If the student answers incorrectly, the system guides the student into obtaining the right answer. The Willow system [14] requires several unmarked reference answers for each question. It also uses LSA to evaluate students' answers written in English or Spanish. In a computer science course, it achieved on average 0.54 correlation with the teacher's grading. A system currently in use at the University of Witwatersrand [15] uses LSA and clustering techniques. It achieves between 0.80 and 0.95 correlation with the teacher's grading.

3 Methodology

In this section, we briefly present the work on OeLE [2] and how we have adapted it and expanded on it in our system. Our focus has been on grading students' answers to questions in a computer algorithms course taught in French.

3.1 Natural Language Processing

For each of the online exam's questions in OeLE [2], the ideal answer provided by the teacher and the students' answers are processed similarly. The GATE software performs most of the NLP tasks, and the Protégé software is used to build the course ontology and encode it in OWL. While OeLE is written in Java and uses the Jena framework to process the encoded ontology, our system is done in PHP and we developed our own ontology-processing code. It is important to note that OeLE and our system use OWL for knowledge representation, but do not utilize its inference services. In this paper, we use the same terminology as [2]. We refer to OWL classes as *concepts*, to object properties as *relations*, and to data properties as *attributes*. Also, *entity* is used as a generic term for concept, relation, or attribute, while *property* is used for relation or attribute.

The NLP consists of three phases: *Preparation*, *Search*, and *Set in a context*. The *Preparation* phase consists of spell-checking, sentence detection, tokenization and POS tagging. In the *Search* phase, the linguistic expressions are detected and matched against the course ontology. Finally, the *Set in a context* phase associates the attrib-

utes and values to their respective concept, and also identifies which concepts participate in a relation.

In OeLE, the texts are annotated *semiautomatically*, meaning that the teacher only needs to manually annotate the fragments unknown to the system or incorrectly tagged. In our system, the natural language processing is done manually for the moment, as GATE does not sufficiently support French (out-of-the-box) for our purposes. Performing automatic French annotation is planned as a future work.

As an example, we use an actual question from a computer algorithms course given at our university: "Describe Depth-First Search (DFS)". Table 1 shows the annotation set (at the end of the NLP phase) of the partial student's answer: "Depth-First Search (DFS) is an exhaustive algorithm that explores a graph..." The ideal answer supplied by the teacher is similarly annotated; however, for every annotated entity, a numerical value ought to be supplied specifying the relative importance of that entity within the question.

Table 1. Example annotated answer of a student to describe DFS.

Category	Description
Concept	DepthFirstSearch
Concept	Algorithm
Concept	Graph
Attribute	Exhaustive
Relation	IsA
Relation	Explores

3.2 Conceptual Grading

The grading stage consists of calculating the semantic distance between the annotation sets (obtained in Section 3.1) of each student's answer and that of the teacher's ideal answer, with respect to the course ontology. Because of space limitations, we cannot give detailed calculations for the example. The reader is advised to see the full explanation in the original publication [2], or an easy-to-follow example in [16].

The formulas used in [2] for calculating the semantic distances are given below. In every function, teacher-provided constants allow for certain elements to be weighted more or less heavily according to their importance. The best combination of these constants is problem-dependent and should be discovered empirically. The "linguistic distance" between the textual representation of the entities in the student and teacher's answer is also taken into account. All functions return values in the [0,1] interval.

Concept similarity. To calculate the concept similarity (*CS*) between concepts c_i and c_j, the following function is used:

$$CS(c_i, c_j) = cp_1 \times CP(c_i, c_j) + cp_2 \times PS(c_i, c_j) + cp_3 \times EQ(c_i, c_j) \quad (1)$$

The constants cp_1, cp_2, cp_3 indicate the relative importance of the corresponding elements. Also, $cp_1 + cp_2 + cp_3 = 1$ and $0 \leq cp_k \leq 1$.

The concept proximity (*CP*) is calculated using the taxonomy formed in the ontology by the class hierarchy defined in OWL. Note that the <*is-a*> relation is explicitly added to the course ontology (with the class as domain and the subclass as image) where `rdfs:subClassOf` is used:

```
<owl:Class rdf:about="DepthFirstSearch">
  <rdfs:subClassOf rdf:resource="Algorithm"/>
</owl:Class>
```

If the concepts c_i and c_j have no taxonomic parent (other than the root), this value is zero, otherwise it is defined as such:

$$CP(c_i,\ c_j) = 1 - \frac{|nodes(c_i, c_j)|}{|concepts|} \tag{2}$$

where $|nodes(c_i,\ c_j)|$ is the number of concepts separating c_i and c_j through the shortest common path through the taxonomic tree, and $|concepts|$ is the total number of concepts in the ontology. A shorter path thus indicates a stronger similarity between the two concepts.

The properties similarity (*PS*) calculates the similarity between the set of properties associated with c_i and c_j. The *properties* of a concept *c* are the union of the set of attributes that have *c* as domain, and the set of relations that have *c* as domain or image.

Lastly, $EQ(c_i,\ c_j)$ uses the Levenshtein distance between the string representation of concepts c_i and c_j, written $L(c_i,\ c_j)$ below, and is defined as follows:

$$EQ(c_i,\ c_j) = \frac{1}{1+L(c_i,\ c_j)} \tag{3}$$

Attribute similarity. The attribute similarity between two attributes a_i and a_j of two concepts is calculated by a similar function:

$$AS(a_i,\ a_j) = at_1 \times EQ(a_i,\ a_j) \\ +at_2 \times VS(a_i,\ a_j) + at_3 \times CS(conc(a_i),\ conc(a_j)) \tag{4}$$

Here also, the non-negative constants at_1, at_2, at_3 must add up to 1. The function $conc(a)$ returns the (most specific) concept which is in the domain of *a*. The function $VS(a_i,\ a_j)$ is defined as such:

$$VS(a_i,\ a_j) = \frac{|vals(a_i) \cap vals(a_j)|}{|min_{k=i,j}\{|vals(a_k)|\}|} \tag{5}$$

that is, the similarity of their value sets. The function $vals(a)$ returns the image of the attribute *a*.

Relation similarity. The relation similarity between two relations r_i and r_j is calculated in a similar manner:

$$RS(r_i, r_j) = rl_1 \times EQ(r_i, r_j)$$
$$+rl_2 \times CS(dconc(r_i), dconc(r_j)) \times CS(dconc(r_i), dconc(r_j)) \quad (6)$$

It is required that the sum of the non-negative constants rl_1, rl_2 be 1. The function $dconc(r)$ returns the most specific concept in the domain of *r*, while $iconc(r)$ returns the most specific concept in the image of *r*. The concept similarity is calculated twice, to compare the domains of the relations r_i and r_j (obtained by $dconc(r)$) and the images of the relations (obtained by $iconc(r)$), respectively.

Global evaluation. In order to accomplish the evaluation of a question, each of the concepts of the student's answer is associated with the closest concept of the ideal answer, given that each concept can only be used once. The similarity between each pair of concepts is then calculated and is multiplied by the relative numerical value of the concept in the ideal answer. The similarity is then added to the final grade. The same process is repeated for relations and attributes.

3.3 Procedural Knowledge Grading

Our system uses the same grading algorithm as OeLE [2]. The students' answers are compared to the teacher's ideal answer. The grades are calculated based on the most similar entity in the expected answer. In OeLE, the order of the entities is not factored in the grade and any permutation of the linguistic expressions of the student's answer yields the same grade.

However, this is not appropriate for assessing procedural knowledge in our system. If the above method is applied to evaluate text describing procedural knowledge such as algorithms-related answers, the grade calculation ought to take into account the relative order of a subset of concepts expressing procedural knowledge.

Functional concepts. In order to address this issue, we propose to add *functional concepts* to the course ontology. A functional concept represents a global procedure, a sequence of sub-procedures or individual steps to accomplish a given task.

Let us consider the following example algorithm, *DepthFirstSearch*, given in pseudocode:

```
procedure DepthFirstSearch
  VisitRoot
  VisitFirstChildNode
  VisitOtherSiblings
end
procedure VisitRoot [...]
procedure VisitFirstChildNode [...]
procedure VisitOtherSiblings [...]
```

For every procedure or sub-procedure, we create a corresponding functional concept: *DepthFirstSearch*, *VisitRoot*, *VisitFirstChildNode*, and *VisitOtherSiblings*. The last three sub-procedures could in turn be further decomposed.

The functional concepts allow for a high-level description of the algorithm and mask implementation details, which would be difficult to express in the ontology using relations or attributes. Further decomposition of *VisitRoot* into individual steps could be stated in any of the following ways:

`DepthFirstSearch <visits> Root` [using relation <visits>]
`VisitRoot <visits> Root` [same relation with a more specific concept]
`Root.visited=true` [the value of the attribute <visited> *becomes* true]

Representing functional concepts in OWL. Relationships between functions are defined as meta-functions in [17]. These meta-functions are implemented in our system as relations between two functional concepts. In this example, two instances of the <*is-preceded-by*> relation are needed. One instance is needed between *VisitFirstChildNode* and *VisitRoot*, because the root has to be visited first, and another between *VisitOtherSiblings* and *VisitFirstChildNode*, because the first child node should be visited first. Similarly, three instances of the <*is-achieved-by*> relation are used between *VisitRoot* and each of the remaining functional concepts.

The same idea is found in [18], where the relation *preceded_by* is defined similarly to <*is-preceded-by*> and can be used to order any pair of classes *P* and P_1. In other words, *P preceded_by* P_1 is defined as "Every *P* is such that there is some earlier P_1". This relation is defined as transitive, and is neither symmetric, reflexive nor antisymmetric.

In [19], an irreflexive and transitive relation *precedes* is used when "the sequence of the related events is of utmost importance for the correct interpretation". This paper also defines the inverse relation *follows*.

Similarly, the working draft: "Time Ontology in OWL" [20] of the World Wide Web Consortium (W3C) states that: "There is a *before* relation on temporal entities, which gives directionality to time. If a temporal entity T_1 is before another temporal entity T_2, then the end of T_1 is before the beginning of T_2." This relation is part of the *time* namespace.

In our implementation, the functional concepts and the <*is-preceded-by*> relation are defined as such in OWL:

```
<owl:Class rdf:about="FunctionalConcept"/>
<owl:Class rdf:about="DepthFirstSearch">
  <rdfs:subClassOf rdf:resource="FunctionalConcept"/>
</owl:Class>
<owl:Class rdf:about="VisitRoot">
  <rdfs:subClassOf rdf:resource="DepthFirstSearch"/>
</owl:Class>
<owl:Class rdf:about="VisitFirstChildNode">
  <rdfs:subClassOf rdf:resource="DepthFirstSearch"/>
</owl:Class>
```

```
<owl:Class rdf:about="VisitOtherSiblings">
  <rdfs:subClassOf rdf:resource="DepthFirstSearch"/>
</owl:Class>
<owl:ObjectProperty rdf:about="IsPrecededBy"/>
```

Note that the <*is-achieved-by*> relation is implied by the class hierarchy rooted at the concept *FunctionalConcept*, just as the <*is-a*> relation is implied by the class hierarchy in OeLE.

For every algorithm, a separate (meta) ontology lists the required orderings specific to that algorithm. Although there exists many algorithms for graph exploration, we only need to define the functional concepts once in the course ontology, and their ordering can then be declared in a separate ontology. For instance, the *Breadth-FirstSearch* algorithm can be defined with the same functional concepts as above, only ordered differently.

For *DepthFirstSearch*, the meta-ontology is as follows:

```
VisitFirstChildNode <is-preceded-by> VisitRoot
VisitOtherSiblings <is-preceded-by> VisitFirstChildNode
```

Note that the following relation is also inferred by the transitive property:

```
VisitOtherSiblings <is-preceded-by> VisitRoot
```

Grading with functional concepts. In our approach, the question evaluation process remains mostly unchanged. No special treatment is given to the functional concept class hierarchy rooted at the concept *FunctionalConcept*, even though its implied relation is <*is-achieved-by*>, rather than the <*is-a*> relation implied for the other concepts. This takes into account function nesting and composition, while allowing calculating the proximity of the functional concepts.

However, the global evaluation of a student answer *R* takes into account the algorithm-specific orderings of the meta-ontology. The new evaluation function is given below:

$$FG(R) = GE(R) \times (1 - od(1 - DF(R))) \tag{7}$$

The final grade (*FG*) for the student answer *R* is proportional to the global evaluation of the answer, $GE(R)$, obtained from Section 3.2. Here, od is a constant in the interval [0,1] allowing the teacher to adjust the relative importance of the correct ordering of concepts in the global evaluation. The ordering factor of the answer, $DF(R)$, is defined as follows:

$$DF(R) = \frac{DD(R)}{|orderings|} \tag{8}$$

where $DD(R)$ represents the number of functional concepts having the right ordering in the student answer *R*, and $|orderings|$ the number of functional concepts orderings in the meta-ontology.

It should be noted that if functional concepts in the student's answer are ordered with the opposite relation (that is, <*is-followed-by*>), the evaluation algorithm inverts the relation between the functional concepts.

Also, the individual student grades are affected by the number of defined orderings. If there are only a few orderings, as demonstrated below, students are strongly penalized for every mistake. This is also the case with the concept proximity defined in Formula 2, where the number of concepts in the ontology affects students' grades. However, we can assume that the course ontology is fixed during evaluation, and that the students' grades are therefore affected similarly (in a linear fashion).

4 Working Example and Results

Using Depth-First Search as an example, we can quantify the effect of the new evaluation function on a student's answer. To simplify, we omit the conceptual grading of the answer and concentrate on the functional grading. Since the same entities are present in both the student and teacher's answers, the conceptual grade is 100%. The ideal functional answer could be as follows: "Depth-First Search **first** visits the root [of a graph], then [recursively] visits its first child node **before** visiting its other siblings." Table 2 shows the produced functional concepts.

Table 2. Example annotation of ideal answer to describe DFS (using only functional concepts).

Category	Description
(Functional) Concept	DepthFirstSearch
(Functional) Concept	VisitRoot
(Functional) Concept	VisitFirstChildNode
(Functional) Concept	VisitOtherSiblings

Any permutation of this ideal answer taken as input by the original approach would yield a grade of 100%. Now, consider the following student's answer: "Depth-First Search visits the root [of a graph], then [recursively] visits its first child node **after** visiting its other siblings." Here, "after" inverts the ordering of the two last concepts (highlighted in bold below), yielding the following answer:

Table 3. Example annotation of student's answer for DFS (using only functional concepts).

Category	Description
(Functional) Concept	DepthFirstSearch
(Functional) Concept	VisitRoot
(Functional) Concept	**VisitOtherSiblings**
(Functional) Concept	**VisitFirstChildNode**

The student gave here the incorrect ordering:

```
VisitFirstChildNode <is-preceded-by> VisitOtherSiblings
```

However, these two student orderings are correct:

```
VisitFirstChildNode <is-preceded-by> VisitRoot [inferred]
VisitOtherSiblings <is-preceded-by> VisitRoot
```

As stated above, the conceptual grading of this answer, as performed by OeLE, is 100%. By using the new evaluation function (Formula 7), the final grade (*FG*) becomes:

$$FG(R) = 100\% \times \big(1 - 1.0(1 - 66.67\%)\big) = 66.67\% \quad (9)$$

where the global evaluation (*GE*) is 100%, the ordering factor (*DF*) is 66.67%, and the constant od is given a value of 1.0. Considering that the ideal answer to this algorithm contains only three orderings for pairs of functional concepts (one is inferred) and that a third is out of order, this low grade seems acceptable, or at least a reasonable improvement over the former grade of 100% that would have been attributed had we only used the conceptual grading system.

5 Conclusion and Future Work

The work presented in this paper adapts the OeLE system to include procedural knowledge. The example was taken from an algorithms course given at Université de Moncton. This approach could be used in other domains where procedural knowledge is central to processing the text. For example, [18] and [19] apply similar methods to biomedical ontologies.

The approach put forth in this paper introduces functional concepts to represent procedural knowledge in ontologies. The class hierarchy of functional concepts is considered as a series of instances of the relation <*is-achieved-by*> instead of <*is-a*>. For every computer algorithm (or procedure, for other domains), a series of instances of the relation <*is-preceded-by*> specify an ordering for pairs of functional concepts.

In this paper, the texts were annotated manually. We are considering annotating the French texts semiautomatically as future work. The detection of the orderings (detecting keywords such as “first”, “before”, “after” in the example of Section 4) could also be performed automatically.

In the case where the student answer uses the opposite ordering relation (<*is-followed-by*>), the relation between the functional concepts is inverted prior to evaluation. Some more complex answers could require more inversions, for example if the student wrote “*X* and *Y* should be done after *Z*”.

Future work could also consider flow control structures, such as loops or branches, although the textual representation of those structures without proper indentation or braces could be ambiguous. For example, the *VisitOtherSiblings* functional concept can be decomposed into the following loop: (for every other sibling, *VisitNode*).

Another idea that could be explored would be to add the notion of recursive procedures, such as Depth-First Search. *VisitFirstChildNode* and (every *VisitNode* of) *VisitOtherSiblings* should include recursive calls. As an ideal answer, the teacher could

give either: *DFS.isRecursive=true*, or *VisitFirstChildNode.isRecursive=true* and *VisitOtherSiblings.isRecursive=true*. Depending on the ideal answer given and their own answer, students could be unjustly penalized.

References

1. Bloom, B.S.: Taxonomy of Educational Objectives, Handbook 1: The Cognitive Domain. David McKay Co Inc., New York (1956)
2. Castellanos-Nieves, D., Fernández-Breis, J.T., Valencia-García, R., Martínez-Béjar, R., Iniesta-Moreno, M.: Semantic web technologies for supporting learning assessment. Information Sciences 181(9), 1517-1537 (2011)
3. Pérez-Marín, D., Pascual-Nieto, I., Rodríguez, P.: Computer-assisted assessment of free-text answers. The Knowledge Engineering Review 24(4), 353-374 (2009)
4. Callear, D., Jerrams-Smith, J., Soh, V.: CAA of short non-MCQ answers. In: Proceedings of the 5th International CAA Conference, Loughborough, UK (2001)
5. Jordan, S., Mitchell, T.: e-Assessment for learning? The potential of short-answer free-text questions with tailored feedback. British Journal of Educational Technology 40(2), 371-385 (2009)
6. Sukkarieh, J., Pulman, S., Raikes, N.: Auto-marking: using computational linguistics to score short, free text responses. In: Proceedings of the 29th IAEA Conference, Philadelphia, USA (2003)
7. Rudner, L. & Liang, T.: Automated essay scoring using Bayes' theorem. In: Proceedings of the Annual Meeting of the National Council on Measurement in Education, New Orleans, LA (2002)
8. Rosé, C., Roque, A., Bhembe, D., VanLehn, K.: A hybrid text classification approach for analysis of student essays. In: Proceedings of the HLT-NAACL Workshop on Educational Applications of NLP, Edmonton, Canada (2003)
9. Mason, O., Grove-Stephenson, I.: Automated free text marking with paperless school. In: Proceedings of the 6th International CAA Conference, Loughborough, UK (2002)
10. Burstein, J., Leacock, C., Swartz, R.: Automated evaluation of essays and short answers. In: Proceedings of the 5th International CAA Conference, Loughborough, UK (2001)
11. Hou, W.-J., Tsao, J.-H., Li, S.-Y., Chen, L.: Automatic Assessment of Students' Free-Text Answers with Support Vector Machines. LNCS 6096, 235-243 (2010)
12. Lutticke, R.: Graphic and NLP Based Assessment of Knowledge about Semantic Networks. In: Proceedings of the Artificial Intelligence in Education conference, IOS Press (2005)
13. Wiemer-Hastings, P., Allbritton, D., Arnott, E.: RMT: A dialog-based research methods tutor with or without a head. In: Proceedings of the 7th International Conference on Intelligent Tutoring Systems, Springer-Verlag, Berlin (2004)
14. Pérez-Marín, D., Alfonseca, E., Rodríguez, P., Pascual-Nieto, I.: Willow: Automatic and adaptive assessment of students free-text answers. In: Proceedings of the 22nd International Conference of the Spanish Society for the Natural Language Processing (SEPLN), Zaragoza, Spain (2006)
15. Klein, R., Kyrilov, A., Tokman, M.: Automated Assessment of Short Free-Text Responses in Computer Science using Latent Semantic Analysis. In: ITiCSE '11 Proceedings of the 16th annual joint conference on Innovation and technology in computer science education, New York, USA, pp. 158-162 (2011)

16. Fernández-Breis, J.T., Valencia-García, R., Cañavate- Cañavate, D., Vivancos-Vicente, P.J., Castellanos-Nieves, D. OeLE: Applying ontologies to support the evaluation of open questions-based tests. In: Proceedings of the KCAP'05 WORKSHOP. SW-EL'05: Aplications of Semantic Web Technologies for E-Learning (in conjunction with 3rd International Conference on Knowledge Capture (KCAP'05)), Banff, Canada (2005)
17. Aroyo, L., Dicheva, D.: Courseware authoring tasks ontology. In: Proceedings of the International Conference on Computers in Education, pp. 1319-1320. (2002)
18. Smith, B., Ceusters W., Klagges, B., Köhler, J., et al.: Relations in biomedical ontologies. Genome Biology 6(R46) (2005)
19. Schulz, S., Markó, K., Suntisrivaraporn, B. Formal representation of complex SNOMED CT expressions. BMC Medical Informatics and Decision Making 8(1) (2008)
20. World Wide Web Consortium (W3C), http://www.w3.org/TR/2006/WD-owl-time-20060927/, last accessed 2012-11-21.

Short paper: Using Formal Ontologies in the Development of Countermeasures for Military Aircraft

Nelia Lombard[1,2], Aurona Gerber[2,3], and Alta van der Merwe[3]

[1] DPSS, CSIR
nlombard@csir.co.za
http://www.csir.co.za
[2] CAIR - Centre for AI Research
Meraka CSIR and Univerity of Kwazulu-Natal
http://www.cair.za.net/
[3] Department of Informatics
University of Pretoria, Pretoria
http://www.up.ac.za/
South-Africa

Abstract. This paper describes the development of an ontology for use in a military simulation system. Within the military, aircraft represent a significant investment and these valuable assets need to be protected against various threats. An example of such a threat is shoulder-launched missiles. Such missiles are portable, easy to use and unfortunately, relatively easy to acquire. In order to counter missile attacks, countermeasures are deployed on the aircraft. Such countermeasures are developed, evaluated and deployed with the assistance of modelling and simulation systems. One such system is the Optronic Scene Simulator, an engineering tool that is able to model and evaluate countermeasures in such a way that the results could be used to make recommendations for successful deployment and use.
The use of formal ontologies is no longer a foreign concept in the support of information systems. To assist with the simulations performed in the Optronic Scene Simulator, an ontology, Simtology, was developed. Simtology supports the system in various ways such as providing a shared vocabulary, improving the understanding of the concepts in the environment and adding value by providing functionality that improves integration between system components.

Keywords: Ontology, Countermeasure, Simulation, Design Research

1 Introduction

Military forces consider aircraft as important and expensive assets often representing huge investments. The protection of these assets is considered to be a priority by most countries. Protection is needed from various threats and one of

these threats are attacks through enemy missiles such as surface-to-air missiles, which are relatively cheap and easy to operate, and in addition, widely available in current and old war-zone areas [1]. These surface-to-air missiles are often complex and they are continually being updated to withstand aircraft countermeasures. In addition, missile systems differ from one another and the need to understand how each type of missile reacts in an aircraft engagement is crucial in the development of aircraft protection countermeasures[1]. The development of these countermeasures is often not possible in real-life situations, and modelling and simulation are therefore necessary for the development of aircraft protection countermeasures. Figure 1 illustrates a military aircraft ejecting a flare, which is a specific type of countermeasure used to protect against missile attacks.

Fig. 1. Countermeasures are implemented on aircraft to protect against missile attacks. Aircraft Ejecting Countermeasures Flares (www.aerospaceweb.org)

Simulation systems model real-world objects and simulate them in an artificial world [2]. One such a simulation system is the Optronic Scene Simulator (OSSIM), which has an application called the Countermeasure Simulation System (CmSim). CmSim uses models of real world objects such as the aircraft and the missile, and simulates different scenarios to evaluate the behaviour of these models under different circumstances [2]. Often these evaluations require substantial computing power and it is not uncommon to wait a few hours for simulation results.

At present, various problems are experienced when constructing the input models for CmSim simulations. Because models are used as input into CmSim simulations, it is necessary to ensure that these models are adequate and accurate for useful simulations. The input model is adequate when it captures sufficient input variables and context, and a model is accurate when it correctly captures the input variables and relations. It is for example possible to create input models that are syntactically correct, but the interaction between the models are not correctly set up in the simulation and therefore the results have no correlation with the real world. In addition, different users with various roles work with the system and it is necessary to acquire a common understanding and vocabulary for the constructs of the models and their characteristics. Furthermore, the cre-

ation of reference models for reuse across the user base would ensure better use of resources and time.

When investigating possible technologies that support modelling within information systems, ontologies and ontology technologies feature extensively. One of the original definitions for the term *ontology* is that by Gruber who defined an ontology as a formalisation of a shared conceptualisation [3]. A formal conceptualisation is a representation in a formal language of the concepts in a specific domain representing a part of the world. Formal ontologies are therefore ontologies constructed using a formal representation language such as Description Logics (DL) [4][4]. *Ontology* is also used as a technical term denoting an artefact that is designed for the specific purpose of modelling knowledge about some domain of interest. Typically a domain ontology provides a *shared and common* understanding of the knowledge in the chosen domain [5]. Given the characteristics and purpose of ontologies, we decided to investigate the use of an ontology to address the identified needs when constructing CmSim Models.

The remainder of this paper is structured as follow: Section 2 provides some background of the simulation environment and why it was necessary to build an ontology, as well as some background on ontologies. Section 3 discusses the development and nature of Simtology. Sections 4 and 5 discuss the contribution and conclude the paper in addition to discussing future work, as well as possible extensions to the ontology.

2 Background

One of the largest investments in the military of a country is aircraft. Aircraft is the target of unfriendly forces in order to weaken the military forces of a country. These attacks include attacks executed by shoulder-launched missiles, which are portable, easy to use and relatively easy to acquire. In order to counter these missile attacks, the military deploy various kinds of countermeasures on aircraft, and these countermeasures are developed, evaluated and deployed with the assistance of modelling and simulation systems such as the Optronic Scene Simulator (OSSIM).

2.1 The Simulation System Environment

CmSim is a software application that is part of the broader Optronic System Simulation (OSSIM) system [2]. OSSIM is an engineering tool used to model and evaluate the imaging and dynamic performance of electro-optical systems under diverse environmental conditions. OSSIM are typically used for the following applications:

- Development of optronic systems
- Mission preparation

[4] For the remainder of this paper we mean *formal ontology* when we use the term *ontology*

– Real-time rendering of infra-red and visible scenes

CmSim is specifically designed to do countermeasure evaluation for the protection of military aircraft. Models of the aircraft, the missile, the countermeasure and the environment are used to construct a scenario that simulates what will happen in the real world [2]. The models are used as input into CmSim, and it is necessary to carefully construct these models to get accurate simulations results. The generation of simulation results are complex and time consuming, and when inaccurate or faulty input models are used, valuable time is lost.

In order to construct better input models, it is necessary to improve the understanding of the simulation and the meaning of concepts in the simulation environment. Users of models often does not know what models exist already, to what level the models were constructed, and the scenarios that might be possible in the simulation, and knowledge is not shared between different role-players. The simulation practitioner setting up the simulation scenario might not have specialist knowledge of how the models interact, and can set up scenarios that are syntactically correct but do not correlate with the real world scenario. There is therefore a need to capture the specialised knowledge in reference models that could be used before the scenario is fed to the simulation. Figure 2 depicts the different role-players that could be involved in the simulation environment.

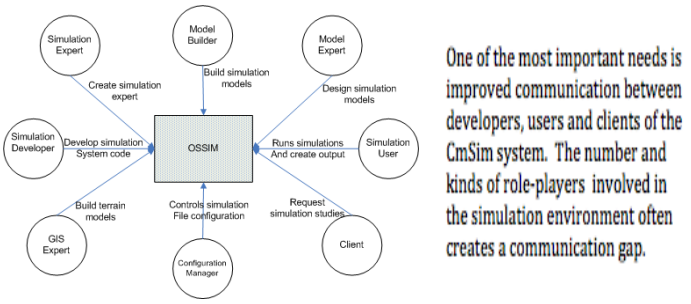

Fig. 2. Different Role-players Involved in a Simulation Environment

In order to address the above mentioned needs, we initiated a project based following the guidelines of design science research (DSR) [6]. DSR provides a research method for research that is concerned with the design of an artefact that solves an identified problem. The creation of an ontology based application was identified as a possible solution to the needs articulated when constructing OSSIM simulation models. DSR will be described further in Section 3.1. The next sections briefly introduce background on ontologies in computing.

2.2 Ontologies and Ontology Tools

The origins of the term *ontology* could be traced to the philosophers of the ancient world who analysed objects in the world and study their existence [3]. Modern ontologies use the principles of the ontology of early philosophers [7]. Ontologies formally describe concepts, so it is often used to capture knowledge of a specific domain. The role of ontologies in a specific domain are thus generally defined by [5] as to:

- Provide people and agents in a domain with a shared, common understanding of the information in the domain;
- Enable reuse of domain knowledge;
- Explicitly publish domain assumptions;
- Provide a way to separate domain knowledge from operational knowledge; and
- Setting a platform for analysis of the domain knowledge.

From the characteristics listed above it is possible to argue that an ontology may be a solution to the problems experienced in OSSIM simulations. Formal ontologies are represented in a specific formal knowledge representation language [4]. For building and maintaining Simtology, we adopted Protégé 4 constructing an OWL ontology. [8, 9]. Protégé is widely used and support for the use of the editor and the development of ontologies are readily available [10–12]. Protégé bundles reasoners such as Fact++ and Pellet with the environment [9, 13] and we used these reasoners to test for consistency or to compute consequences over the knowledge base during the development of Simtology [14].

2.3 Ontology Use in Modelling, Simulation and Military Systems

Within computing, modelling and simulation are used to build a representation of the real world and simulate the behaviour of objects presented in the models [2]. A simulation system is a specific application that uses a model as input and execute a computer program that determines consequences and resulting scenario information about the system being investigated [15].

Military systems and the knowledge captured therein are complex and often consist of layered information from different sources. To support this view, Clausewitz, in his book, *On War*, wrote about military information as follow [16]:

> '...three quarters of the information upon which all actions in War are based on are lying in a fog of uncertainty...'
> '...in war more than any other subject we must begin by looking at the nature of the whole; for here more than elsewhere the part and the whole must always be thought of together...'

Furthermore, Mandrick discussed the use of ontologies to model information in the military environment. According to Mandrick, ontologies in the military

must adhere to the same requirements as ontologies in other domains, as described in Section 2.2. Important aspects to highlight is the ability of the ontology to provide a common vocabulary between planners, operators and commanders in the different military communities [16].

At present the adoption of ontologies in the military domain is primarily for support of command and control in the battlefield, as well as the management of assets and the sharing of knowledge[11, 17]. We also find ontologies where there is a need to integrate different data sources and the communication between these data sources [18, 19]. Although ontologies are used in the military modelling and simulation domain, examples are sparse and at present do not support the construction of input models for systems such as OSSIM. It could be argued that Simtology will therefore present a unique contribution to military information management.

3 Simtology

The development of Simtology was in response to the identified needs when using the Optronic Scene Simulator (OSSIM) [2] to develop countermeasures for missile attacks on aircraft as discussed in Section 2.1.

3.1 The Design and Development Process

The research design adopted for the development of Simtology, was Design Research (DSR). DSR is a research methodology for the design and construction of computing artefacts through the use of *rigour* (the use of fundamental knowledge) and *relevance* (basing the development of the artefact on real needs) [6, 20]. In this project, the artefact is Simtology, the fundamental knowledge is obtained from ontology knowledge, and the need is the construction of models for the OSSIM simulation environment. A DSR execution method that was proposed by Vaishnavi et al.[21] is depicted on the left in Figure 3. This method was adopted for this research project, and the development of Simtology is discussed further according to the steps in Figure 3.

3.2 Awareness and Suggestion

The first steps in the design research process is *awareness of the problem* and *proposing possible suggestions* for a solution. The following list summaries the issues and needs in the simulation system as discussed in Section 2.1 that created *awareness of the problem*:

- Different role-players: There are developers, model builders and users involved in the system. Inconsistencies in the terminology used between different users often led to frustration and wrong use of concepts. There is lack of a common vocabulary that is shared by everyone involved in building and using the system.

- Model complexity: One of the main characteristics is the ability of the system to execute models at different levels of detail. This poses a problem to users, when to know at which level of detail a model is implemented at.
- Reference models: Specific users that only interact with the system at a certain level, need more technical insight into model detail to know what is available in the system. This means that reference models are required that can define domain-specific concepts to these users.
- Model Interaction: A simulation consists of a scenario that is built up from interacting models. The models interact using a set of rules but there is currently no rules that verify model behaviour when a scenario are constructed.

Previous research efforts in the simulation environment attempted to address the the need for a standard notation for documentation of the simulation models. This proved to be problematic and one of the suggestions for further research was to investigate the use ontologies in the simulation environment. The *suggestion* according to the DSR process is therefore that a formal ontology is created to address the above mentioned needs for the simulation environment.

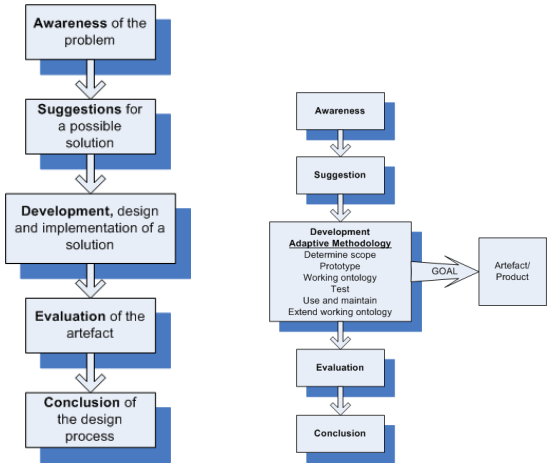

Fig. 3. The Design Research Process on the left, and the Adaptive Methodology Process on the right

3.3 Development

The ontology was build using the approach followed by the researchers who develop the Adaptive Methodology [22], and was chosen for its lightweight, incremental approach. Figure 3 depicts the development process steps as well as the alignment with the design process.

- **Scope and Purpose:** The first step is to scope the purpose and the extent of the ontology. In the case of a domain ontology, the concepts of the domain must be included. It is not necessary to include all the concepts of the domain. The level of detail will be determined by the purpose of the ontology.
- **The Use of Existing Structures:** There are several documents, structures and sources available in the OSSIM simulation environment available to use in order to gather information to develop the ontology and to, for example, make a list of the concepts in the simulation. Modelling reports, installation guides, white papers and technical documentation, as well as the source code of the system and the documented test point configurations were used as input into the ontology development process.
- **The Prototype:** The prototype structure is the first version of the ontology that is operational. The prototype for the simulation environment contains only a selected set of components from the domain. The concepts are on a high level and the nested structures of complex concepts were not included in the prototype. The prototype was developed in Protégé and is illustrated in Figure 4 on the left.
  The prototype is a proof-of-concept and in this project it played an important role to demonstrate the feasibility of the suggested solution. The prototype ontology supported the role of an ontology in the simulation system environment, and supported an ontology as a solution to a shared, common vocabulary. The tools also provided graphical views of the concepts defined in the ontology.
- **Development of the Working Ontology:** During this phase the prototype ontology was expanded by adding concepts from the domain not previously included in the ontology, as well as developing new functionality. The working ontology contains a full set of domain concepts that describe the simulation models and model properties and is called Simtology. The next section describes Simtology in more detail, as well as how Simtology is used in CmSim.

3.4 Description of Simtology

To do a simulation in CmSim, a scenario must be set up to act as input to the simulation. The scenario consists of various configuration files but the main file is the scenario file itself, which contains links to all other files necessary to describe a scenario and the components in it. Although the prototype already contained enough information to set up basic scenarios, Simtology contains all the concepts in the domain of CmSim.

The first task was thus to expand the prototype to present not only the basic objects, but all the possible objects in the CmSim domain. The classes and properties were expanded. The following list describes the concepts and properties defined in Simtology.

- **Concepts:** In Simtology, an example of a concept representing all the individual aircrafts modelled in the simulation environment, is `Aircraft`. Figure

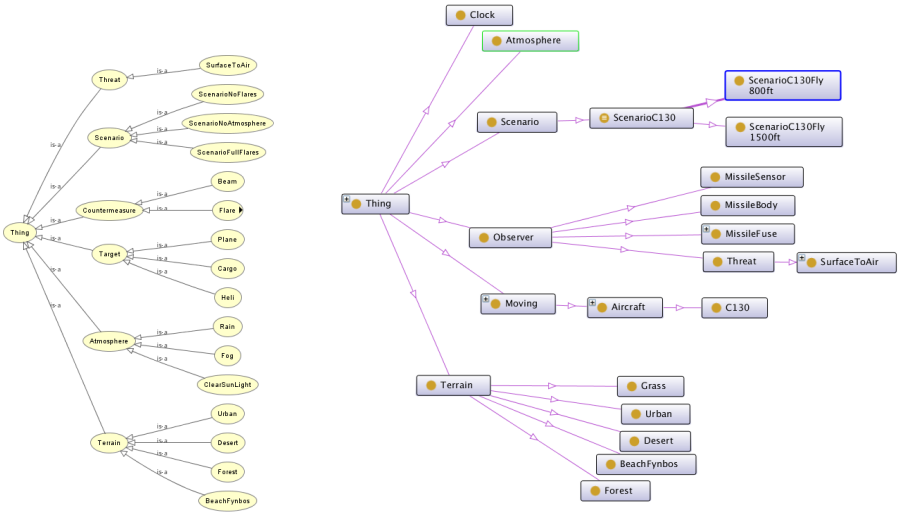

Fig. 4. Concepts from the prototype ontology on the left, and concepts in the final version of Simtology on the right

4 depicts an extract of the top-level concepts defined in Simtology, where the concepts were selected to present those in a simulation scenario. The main concepts in Simtology are the Moving, Observer and Scenario concepts. The choice of concepts relied heavily on the structure of the simulation configuration files. Therefore objects of type Moving have specific behaviour in the simulation and belong together in a concept.

– **Individuals:** Individuals are asserted to be instances of specific concepts. Specific scenarios can be build by choosing individuals from the ontology and thus creating an individual scenario.
`ScenarioC130` is an individual of the `Scenario` concept that uses a specific type of aircraft.

– **Object Properties:** Object properties are used to link individuals to each other. In Simtology, a scenario must have a clock object, so having a clock object is an object property of the scenario concept. The name of the property is "hasClock" and links an individual of class `Clock` to an individual from class `Scenario`.

In Simtology the main object properties belong to a scenario. The following properties are sufficient to denote a valid scenario that can run in a simulation: `ScenarioC130` hasClock `Clock10ms`
`ScenarioC130` hasTerrain `TerrainBeachFynbos`
`ScenarioC130` hasMoving `C130Flying120knots`
`ScenarioC130` hasObserver `SAMTypeA`
`ScenarioC130` hasAtmosphere `MidLatitudeSummer`

– **Data Properties:** Data properties were added to Simtology to add data to individuals. Examples of data properties are geometric locations of moving objects, or data belonging to the class `Clock`, as depicted below:
`Clock10ms` hasInterval `10ms` and `Clock10ms` hasStopTime `10sec`

Functionality: A scenario can be complex and rules were built in to ensure that a valid scenario is constructed, for instance only certain types of flares can be used as a valid countermeasure on a specific aircraft. After building the scenario in the ontology, the scenario can be processed by a reasoner. The reasoners are used to compute the inferred ontology class hierarchy and to do consistency checking after a scenario is created.

Additional functionalities were developed for use with Simtology such as the integration of the ontology with the graphical user interface (GUI) used to set up the simulation. The ontology is used to populate the elements in the interface, resulting in several advantages such as that only one source of simulation information has to be maintained, as well as that the ontology can be used to change the language displayed in the GUI .

Functionalities were also developed to write out scenarios created in the ontology to files that can act as input to the simulation. This made it possible that a scenario can first be checked for logical correctness before it is run in the simulation. Modelling errors not handled by the simulation software are handled early in the simulation process by using the reasoning technology in the ontology. By having a scenario defined in the ontology, it is possible to export a high-level description of a scenario and its components to be used for reporting and documentation of simulation studies.

Testing of Simtology Testing the ontology was an important step towards creating a useful Simtology. When an ontology is small with a few concepts, it is easier to identify modelling problems but when there are large numbers of concepts with complex relationships, it is important to test the ontology regularly in order to avoid inconsistencies immediately and eliminate rework. During ontology verification the focus was mainly to ensure that the ontology was built correctly and that the ontology concepts match the domain it represents. The test phase of the ontology is part of the adaptive methodology process and this phase complements the evaluation phase of the design research process.

4 Evaluation

In Section 3.2, the simulation system environment was discussed. In order to evaluate the use of Simtology in the simulation system and the contribution it has for the improvement of the environment, the issues mentioned in Section 3.2 are used as evaluation criteria. The identified issues are 1) different users; 2) model complexity; 3) reference models; and 4) model interaction. When evaluated against the identified issues, Simtology provided the necessary solutions.

- **Different users:** Simtology provided a common, shared 'language' to assist with eliminating ambiguities and the inconsistent use of terminology by the different users of the system. The feedback by all concerned users was positive. An example of how Simtology assisted with regards to a common understanding is in the use of ambiguous terms. Some terms in the simulation had different meanings depending on the user using it and the application it was used for. An example of such a term is *countermeasure*, which was vague and previously many different types of countermeasures existed. In Simtology the concept `Countermeasure` was defined in such a way that it can be used as an explanatory tool to illustrate the different countermeasures available in the simulation as well as the use of each countermeasure.

  The Protégé editor allows for several ways to communicate the ontology, for example a graphical display of the concepts and the relations in the ontology A visual display of the different components in the simulation leads to better communication between all the people involved.
- **Model complexity:** Simtology formally defined the concepts, properties and individuals necessary for the construction of input models. When a user uses Simtology to construct her input model, the appropriate level of detail and complexity of the input models are specified.
- **Reference models:** Simtology provides a reference model for all the different users of the system to create their specific input models from. After introducing Simtology, very few problems were experienced by users when constructing simulation models because Simtology acted as a reference model informed their specific model design.
- **Model Interaction:** A simulation consists of a scenario that is build up from interacting models. Simtology provides a common shared language to be used in the simulation environment for both users and when interacting with other applications. The definitions of concepts in the system are kept in Simtology and made available to applications in the environment such as the Graphical User Interface.

  As a final evaluation, the guidelines proposed by Hevner et al. [6] for a design research artefact were used to evaluate and present the research process followed to develop Simtology. This discussion is outside the scope of this paper but it was demonstrated that the construction of Simtology followed the proposed guidelines.

5 Conclusion and Future Work

The outcome of the research project was Simtology, a domain ontology for the simulation environment that contains the model information for simulation scenarios. An added benefit was that the process to analyse the contents of the simulation environment to construct the ontology clarified the knowledge in the domain.

During the construction of Simtology, the following observations were made:

- With regards to modelling, it is important to distinguish part-of from subclass-of. An aircraft body is part of an aircraft, not part of a specific type of aircraft.
- It is important to correctly model roles. Modelling a missile as an *observer* in the simulation means that it can never be used in the simulation as an *object of type moving*. In Simtology, a missile can therefore never be used in a different role.
- Another important modelling decision has to do with the modelling of individuals vs. concepts. This decision has an impact on how the ontology could ultimately be used. The choice between concept and individual is often contextual and application-dependent but it needs to be evaluated in one of the development cycles.
- The development and use of the ontology should be an iterative process. As new functionality is added, it must be tested, used and evaluated. Existing functionality is maintained by making changes where necessary. Proper version control is therefore also necessary when constructing ontologies.

Several advantages of having an ontology in the simulation environment emerged after the ontology was created. The ontology can, for instance, be used in training exercises to show aircraft personnel the technical detail of the countermeasures deployed on the aircraft. Furthermore, the simulation environment is always expanding and improving through the addition of new models, the addition of new properties to existing entities in the system or through the addition of new functionality to entities. Future versions of the ontology need to incorporate these changes and there should therefore always be future expansions to the Simtology ontology. Furthermore, Simtology should ideally be expanded to not only include concepts in CmSim, but also in the Optronic Scene Simulator. One of the planned functions to be developed is to reverse engineer previously run simulations and add the scenario descriptions of those simulations to the ontology.

References

1. Birchenall, R.P., Richardson, M.A., Butters, B., Walmsley, R.: Modelling an infrared man portable air defence system. Infrared Physics & Technology (July 2010)
2. Willers, C., Willers, M.: Ossim: Optronics scene simulator white paper. Technical report, Council for Scientific and Industrial Research (2011)
3. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (1993) 199–220
4. Baader, F., Horrocks, I., Sattler, U.: Description logics as ontology languages for the semantic web. In Hutter, D., Stephan, W., eds.: Mechanizing Mathematical Reasoning: Essays in Honor of Jörg H. Siekmann on the Occasion of His 60th Birthday. Volume 2605 of Lecture Notes in Artificial Intelligence. Springer-Verlag (2005) 228–248
5. Noy, N.F., McGuiness, D.L.: Ontology development 101: A guide to creating your first ontology. Technical report, Stanford Knowledge Systems Laboratory (2001)

6. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly **28**(1) (2004) 75–105
7. Guarino, N.: Formal ontology and information systems. In Guarino, N., ed.: Proceedings of the First International Conference on Formal Ontologies in Information Systems (FOIS-98), June 6-8, 1998, Trento, Italy, IOS Press, Amsterdam, The Netherlands (1998) 3–15
8. OWL: Owl2 web ontology language. Available at http://www.w3.org/TR/owl2-overview. [1 April 2011]
9. Protege: The protege ontology editor. Available at http://protege.stanford.edu. [13 April 2011]
10. Gáevic, D., Djuric, D., Devedíc, V.: Model Driven Engineering and Ontology Development. 2. edn. Springer, Berlin (2009)
11. Schlenoff, C., Washington, R., Barbera, T.: An intelligent ground vehicle ontology to enable multi-agent system integration. In: Integration of Knowledge Intensive Multi-Agent Systems. (2005) 169–174
12. Nagle, J.A., Richmond, P.W., Blais, C.L., Goerger, N.C., Kewley, R.H., Burk, R.K.: Using an ontology for entity situational awareness in a simple scenario. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology **5**(2) (2008) 139–158
13. Tsarkov, D., Horrocks, I.: FaCT++ description logic reasoner: System description. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006). Volume 4130 of Lecture Notes in Artificial Intelligence., Springer (2006) 292–297
14. Bock, J., Haase, P., Ji, Q., Volz, R.: Benchmarking OWL Reasoners. CEUR Workshop Proceedings (June 2008)
15. Benjamin, P., Patki, M., Mayer, R.: Using ontologies for simulation modeling. In: Proceedings of the 38th conference on Winter simulation. WSC '06, Winter Simulation Conference (2006) 1151–1159
16. Mandrick, L.B.: Military ontology. http://militaryontology.com/
17. Valente, A., Holmes, D., Alvidrez, F.C.: Using a military information ontology to build semantic architecture models for airspace systems. In: Aerospace Conference, 2005 IEEE. (March 2005) 1–7
18. Winklerova, Z.: Ontological approach to the representation of military knowledge. Technical report, Military Academy in Brno, Command and Staff Faculty, Czech Republic (2003)
19. Smart, P.R., Russell, A., Shadbolt, N.R., Shraefel, M.C., Carr, L.A.: Aktivesa. Comput. J. **50** (November 2007) 703–716
20. Hevner, A., Chatterjee, S. In: Evaluation. Volume 22 of Integrated Series in Information Systems. Springer US (2010) 109–120
21. Vaishnavi, V., Kuechler, W.: Design research in information systems. Available at http://desrist.org/design-research-in-information-systems/ Last Updated 16 August 2009 (January 2004)
22. Bergman, M.: A new methodology for building lightweight, domain ontologies. Available at http://www.mkbergman.com/908/a-new-methodology-for-building-lightweight-domain-ontologies/ (2010)