FREIE UNIVERSITÄT BOZEN
LIBERA UNIVERSITÀ DI BOLZANO
FREE UNIVERSITY OF BOZEN - BOLZANO

KRDB RESEARCH CENTRE
KNOWLEDGE REPRESENTATION MEETS DATABASES

Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy
Tel: +39 04710 16000, fax: +39 04710 16009, http://www.inf.unibz.it/krdb/

KRDB Research Centre Technical Report:

Comprehensiveness versus Scalability: guidelines for choosing an appropriate knowledge representation language for bio-ontologies

C. Maria Keet and Mariano Rodriguez

Affiliation	KRDB Research Centre, Faculty of Computer Science Free University of Bozen-Bolzano Piazza Domenicani 3, 39100 Bolzano, Italy
Corresponding author	Maria Keet keet@inf.unibz.it
Keywords	OWL, DL-Lite, $\mathcal{DLR}$, Description Logics Semantic Web, bio-ontologies
Number	KRDB07-5
Date	May 9, 2007
URL	http://www.inf.unibz.it/krdb/pub/

Comprehensiveness versus Scalability: guidelines for choosing an appropriate knowledge representation language for bio-ontologies

C. Maria Keet and **Mariano Rodríguez**
Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
{keet, mrodriguez}@inf.unibz.it

Abstract

Ontology representation languages for the Semantic Web have their strengths and weaknesses, in particular in the light of deploying them in information systems. We survey and compare the Description Logics-based OWL variants, and the $DL\text{-}Lite$ and $\mathcal{DLR}$ families of languages. We demonstrate distinguishing features with examples from the biological and biomedical domains. Language choices that an ontology developer has to make are, among others, expressivity versus ontology usage for data-intensive tasks and having mereological parthood versus n-ary relations (where $n > 2$). Guidelines are suggested to facilitate choosing the language best fitted for a task.

Introduction

Since the release of the W3C standard of the Semantic Web ontology language OWL in 2004, many bio(medical) ontologies are developed in OWL either *de novo* or have translations from their native language to OWL. An aim is to work toward ontology compatibility to enhance information integration in biomedical domain and to represent formally our understanding of biological and biomedical reality. However, early-adopters from the bio(medical) domain start reporting their first issues with OWL (Bandini & Mosca 2006; Kazic 2006; Marshall *et al.* 2006; Ruttenberg, Rees, & Zucker 2006; Smith *et al.* 2006; Wolstencroft, Stevens, & Haarslev 2007). Their problems concern

I. (perceived) limitations of ontology representation languages for representing biomedical knowledge adequately and contain requirements or proposals for improvements of OWL for biomedicine, and
II. bottlenecks concerning linking data to the ontologies and subsequent performance issues of the software system when performing common reasoning tasks, such as classification and querying.

Applications of biomedical ontologies in the Semantic Web are sparse, but are expected to gain momentum once ontologies can be linked *efficiently* to biological data and used with, *e.g.*, electronic health record management for both annotation and mining hospital information systems, querying whole genomes through an ontology, or even trying to manage the vast amount of metagenomics data (*e.g.*, (Seshadri *et al.* 2007)) through several domain ontologies.

Are Semantic Web technologies and ontology languages anywhere near meeting such goals set by domain experts in biology and biomedicine? A familiar requirement is greater expressivity of the ontology language to enable representing the complexities of biology as comprehensive as possible, which corresponds to type I problems mentioned above. This is being addressed gradually, most notably with the recently proposed OWL 1.1[1]. From the computer science perspective, meeting type II requirements, such as data access through an ontology and ontology-based knowledge- or data integration, however, are somewhat 'behind' compared to expectations of science researchers. One tried and tested solution is the Instance Store (Bechhofer, Horrocks, & Turi 2005; Wolstencroft, Stevens, & Haarslev 2007) that links an expressive OWL ontology to a relational database, but is not scalable to large amounts of data or large ontologies – precisely because of the expressive ontology language used and the optimization algorithms required for automated reasoners, such as Pellet, FaCT, and RACER, to deal with expressive ontology languages. A recently proposed alternative are the so-called 'lite' family of ontology languages (Calvanese *et al.* 2006a; 2006b; 2005), which are less expressive but are better scalable –that is, like one is accustomed to from relational databases – and therefore will be more suitable for use with bio-ontologies in large information systems and across the Semantic Web.

To clarify the differences between these new and extant ontology languages and their performance with intended usages, and, more importantly, the unavoidable trade-offs, we compare 9 Description Logics-based ontology languages and provide an overview of the important distinguishing features and limitations in Section 2. Although one can model freely with as aim to represent a scientific theory as comprehensive and ontologically correct as possible, we focus on the core requirement to actually use ontologies computationally for a range of purposes. Given the identified trade-offs due to expressivity of an ontology language, computational limitations, and (under-)used language features, we suggest guidelines to choose the best suitable formal language for the task at hand (Section 3). Such details for

[1] http://owl1_1.cs.manchester.ac.uk/owl_specification.html (Editor's draft of 27-11-2006).

choosing languages ought to be hidden from the developer, who then could use a user-friendly graphical, diagrammatic, or natural language interface for ontology and software development and usage in an application layer; however, this solution will take more time to realise (although research-grade tools such as iCOM, GrOWL, SWOOP, and Protégé do move in this direction). Therefore, developers have to make informed decisions about the language optimal for the intended purposes of the ontology. With this contribution, we aim to facilitate this process. Conclusions and ongoing research are described in Section 4.

Features and limitations of knowledge representation languages

Knowledge representation languages have their origins in logic and a resulting knowledge base system combines the 'model' (logical theory) with data. This is distinct from ontologies and conceptual models for database development in the sense of both *what* should be represented – to capture a piece of reality versus modelling for a certain application – and *how* it is to be used – 'online' usage with computational support versus 'offline' static representation. Such distinctions are becoming blurred when users require 'ontology languages with conceptual modelling qualities', such as support for data types, and online access to conceptual models for dynamic database connectivity, like for peer-to-peer semantic networks. However, a formal representation *language* can be indifferent about the developers' intentions. Put differently, knowledge representation languages like Description Logics (DL) can be, and are being, used as a unifying paradigm for ontology development and formal conceptual modelling (Baader *et al.* 2003). Therefore, we assess features and limitations of DLs regarding both biomedical ontologies and conceptual models for biological and medical data in the first two subsections, respectively. Afterward, we take inclusion and usage of parthood relations as combined requirement motivated from these subject domains and assess implementation feasibility.

Ontology languages for the Semantic Web

In this subsection, we discuss features and limitations of the expressive OWL languages and alternative DL-based ontology languages of the $DL\text{-}Lite$ family that are more suitable for usage with large amount of data.

OWL features. Within the scope of the Semantic Web for health care and life sciences, biomedical ontologies, ontology representation languages, and formalisms for biomedical data, the focus is on use of the W3C standard Web Ontology Language OWL. "the OWL language" comes in three flavours: *OWL-full* is built on top of RDF, *OWL-DL* is based on the Description Logics (DL) language $\mathcal{SHOIN}$ with additional support for data types, and *OWL-Lite* is based on the DL language $\mathcal{SHIF}$, which is a subset of OWL-DL. OWL-Lite lacks e.g. union and complement of concepts, like that one cannot state that "Apples are *not* Oranges" ($Apple \sqsubseteq \neg Orange$), and has cardinalities restricted to $\geq$ 1, $\leq$ 1, 1 or 0, preventing one to represent, say, that "Benzene consists_of 6 Carbon_atoms". The latter, among other new features, is properly addressed with the next-generation OWL, the draft version OWL 1.1. OWL 1.1 is based on the DL language $\mathcal{SROIQ}$ (Horrocks, Kutz, & Sattler 2006), also has additional support for data types, and extends the functionality of OWL-DL with, a.o., several role properties, such as reflexivity and concatenation, and qualified number restrictions that permits one to specify multiplicity/cardinality also with a qualified role (*i.e.*, the range is defined); more precisely, statements of the types $C \sqsubseteq \geq n\ R.D$ and $C \sqsubseteq \leq n\ R.D$ where n can be any finite integer ≥ 0; hence, now one can include

$$Benzene \sqsubseteq \geq 6\ consists\_of.Carbon\_atom \sqcap \leq 6\ consists\_of.Carbon\_atom.$$

On the other hand, OWL 1.1 is not compatible with RDF, hence, neither with OWL-full. The main differences between the DL-based OWL languages are described by (Cuenca Grau *et al.* 2006) and summarised in Table 1.

Although it may be tempting to choose the language with the greatest expressivity, it comes at a cost: performance of your implementation. Given that bio(medical) ontologies can become quite large and ontology-mediated biological data sets are already large, computational usability of biomedical ontologies turns into an important requirement. We return to this issue in Section 3.

DL-Lite features. $DL\text{-}Lite$ is a family of DL languages whose expressive power is specifically tailored to provide good performance reasoning algorithms in the presence of large amounts data stored in the ABox ('individuals in the ontology') or linked relational databases (Calvanese *et al.* 2006a; 2005; 2006b). Focusing on ontology-based data access and ontology-based database integration, $DL\text{-}Lite$ allows for delegation of data handling to relational databases through database-ontology mappings and algorithms that translate queries posed in terms of a $DL\text{-}Lite$ ontology to suitable queries over the linked database(s).

Modelling features available in the $DL\text{-}Lite$ family – beyond the usual features like DL-concept hierarchies, disjointness between DL-concepts (or roles), and role domain and range specification – are DL-concept and role (relation) value-domains and, implicitly, n-ary relations where $n > 2$; see Table 1 for details. In particular, specifying role values is a novel addition in $DL\text{-}Lite_{\mathcal{A}}$, not available in any other DL language. One can attach an attribute for concrete values of datatypes (e.g, strings, integers, and dates), to a *relation* as well as to a DL-concept (Calvanese *et al.* 2005), thereby allowing the modeller to correctly represent, e.g., the relation between Patient, Hospital and the values for the date of admission, or the region of a gene location on the chromosome *without* having to resort to complicating intermediate reification steps. n-ary relations can be supported through an extension (Calvanese *et al.* 2005) alike in $\mathcal{DLR}$, which does have support for n-ary relations fully integrated in the language. This will be addressed in the next section.

DL languages for formal conceptual modelling

We take a brief look at formal conceptual modelling with DLs, because of the option for common usage of DLs for

Language ⇒ **Feature** ⇓	OWL			*DL-Lite*			$\mathcal{DLR}$		
	Lite	**DL**	**v1.1**	$\mathcal{F}$	$\mathcal{R}$	$\mathcal{A}$	*ifd*	μ	*reg*
Role hierarchy (taxonomy of relations)	+	+	+	-	+	+	+	+	+
N-ary roles (where $n \geq 2$, ternary, quaternary relation etc.)	-	-	-	±	±	±	+	+	+
Role concatenation (limited role composition)	-	-	+	-	-	-	-	-	+
Role acyclicity (least fixpoint construct)	-	-	-	-	-	-	-	+	-
Symmetry	+	+	+	-	+	+	-	-	-
Role values (role attribute values, like strings and integers)	-	-	-	-	-	+	-	-	-
Qualified number restrictions (where the cardinality/multiplicity n may also be ≥ 2)	-	-	+	-	-	-	+	+	+
One-of, enumerated classes (constraining the instances of a class to a pre-defined set of objects)	-	+	+	-	-	-	-	-	-
Functional dependency (UML method, derived-and-stored relation)	+	+	+	+	-	+	+	-	+
Covering constraint over concepts (total/complete covering of the subtypes)	-	+	+	-	-	-	+	+	+
Complement of concepts (disjointness of classes)	-	+	+	+	+	+	+	+	+
Complement of roles (disjointness of roles)	-	-	+	+	+	+	+	+	+
Concept identification (primary key with $>$ attribute)	-	-	-	-	-	-	+	-	-
Range typing (define concept of the 2nd participant in role)	-	+	+	-	+	+	+	+	+
Reflexivity *	-	-	+	-	-	-	-	+	+
Antisymmetry *	-	-	-	-	-	-	-	-	-
Transitivity * ‡	+	+	+	-	-	-	-	+	+
Asymmetry ‡	+	+	+	-	+	+	-	±	-
Irreflexivity ‡	-	-	+	-	-	-	-	+	-

Table 1: Differences between Description Logics-based ontology and conceptual modelling languages; terms in braces are regularly considered as synonyms (for indicative purpose only); indirect or implied support (±); properties of the parthood (*) and proper parthood (‡) relation.

both ontology and conceptual modelling development, the prospect of ontology-driven information systems, database and tool integration through the use of ontologies, and smoothening translation from an ontology to conceptual models and their corresponding databases. The DL $\mathcal{DLR}$ and its extensions were specifically developed to provide a mapping from conceptual modelling languages such as UML, EER, and ORM2 to a DL (Artale & Franconi 1998; Berardi, Calvanese, & De Giacomo 2005; Calvanese, De Giacomo, & Lenzerini 1999; 1998; Keet 2007) and has a mapping to the DIG interface for DL reasoners, such as RACER and Pellet, to enable automated reasoning over conceptual models. This combination is available in the iCOM tool[2], which automatically checks satisfiability of the model (if all DL-concepts can be instantiated), computes derived relations, classifies DL-concepts, and can reason across relations between different conceptual models. Distinct features of $\mathcal{DLR}$s compared to the aforementioned ontology languages are that they all fully support objectification, qualified number restrictions, and n-ary relations where $n \geq 2$. Returning to (Kazic 2006) and (Smith *et al.* 2006) mentioned in the introduction, they noted the OWL shortcoming that it cannot deal with "even simple interactions among pluralities of continuants" (Smith *et al.* 2006). Thus, with $\mathcal{DLR}$ you can represent this. We demonstrate two different examples: A) (Kazic 2006) wants to let thymidine phosphorylase bind with thymidine or phosphate, which are two binary relations with role exclusion, and B) a ternary relation in some biomedical conceptual model for recording epidemiological data on the path of infection of particular HIV subtypes from Donor to Recipient. With the recently mapped ORM2-to-$\mathcal{DLR}_{ifd}$ by (Keet 2007) (ORM2 to OWL-DL is in development), one could avail of the ORM2 diagrammatic representation and, for the more linguistically-oriented Semantic Web users like (Kazic 2006), have it *automatically* verbalized in near-natural language sentences, which is depicted in Figure 1, which was made in NORMA[3]. The corresponding translations into $\mathcal{DLR}_{ifd}$, where the second example involves an automated reification step, is normally hidden from the developer so that a user can focus on the n-ary relation as a whole instead of its cumbersome formalization. For illustrative purpose, the ternary relation R is represented 'behind the scenes' in $\mathcal{DLR}_{ifd}$ as follows:

$$R \sqsubseteq \exists[1]r_1 \sqcap (\leq 1[1]r_1) \sqcap \forall[1](r_1 \Rightarrow (2 : HIVsubtype)) \sqcap \exists[1]r_2 \sqcap (\leq 1[1]r_2) \sqcap \forall[1](r_2 \Rightarrow (2 : Donor)) \sqcap \exists[1]r_3 \sqcap (\leq 1[1]r_3) \sqcap \forall[1](r_3 \Rightarrow (2 : Recipient))$$

and the role exclusion as

$$[binds_T]R_1 \sqsubseteq \neg[binds_P]R_2.$$

$\mathcal{DLR}$s also support primary key identification and func-

[2] http://www.inf.unibz.it/∼franconi/icom.

[3] http://sourceforge.net/projects/orm/.

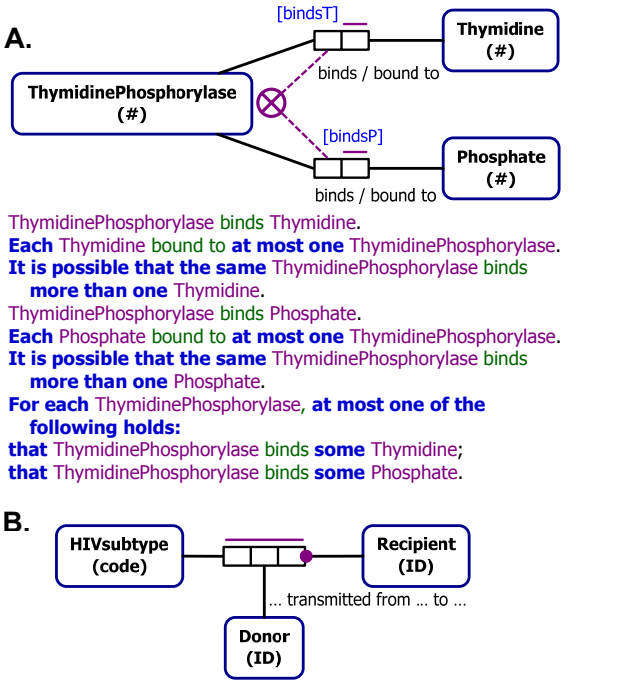

Figure 1: Diagrammatical and textual representation in ORM2 of two examples that have interactions among more than two continuants. A: role exclusion; B: ternary relation.

tional roles for UML methods (in $\mathcal{DLR}_{ifd}$), role acyclicity and transitivity, and role concatenation ($\mathcal{DLR}_{\mu}$, $\mathcal{DLR}_{reg}$), and temporal DL ($\mathcal{DLR}_{\mathcal{US}}$); see Table 1 for details.

An example: representing and using parthood in biology and biomedicine

Part-whole relations are important for representing biological and biomedical knowledge and deriving information of biological or biomedical interest. For instance, it is of biological interest to know if a type of zinc-endopeptidase, *Tetanospasmin*, is always part of all bacteria of type *Clostridium tetani*, and if this is the case, then all humans that are infected with (contain) bacteria of type *C. tetani* will also contain – have as part – *Tetanospasmin* (hence, suffer from the disease tetanus). Even the relatively early efforts in biomedical ontology development ensured inclusion of this relation in one way or another. For instance OpenGALEN has 20 relations categorised as types of part-whole relations[4], it was on a par with the subsumption relation in the Gene Ontology (Gene Ontology Consortium 2004), and the Foundational Model of Anatomy (Rosse & Mejino Jr 2003) has a partonomy firmly integrated in the ontology. More recently, several part-whole relations have been included in the OBO Relation Ontology (RO) (Smith *et al.* 2005), which is used by the Cell Cycle Ontology (Antezana *et al.* 2006). The Cancer Bioinformatics Infrastructure Objects Model (caBIO), on the other hand, struggles with UML's aggregation relation, collections, and DL roles[5], and the Protein Ontology permits, but does not specify semantics of, a parthood relation[6]. From the medical informatics modelling perspective, attempts to solve usage of parthood relations in DL are pursued by (Schulz & Hahn 2004; Schulz, Hahn, & Romacker 2000). These variations in representation of parthood relations are, at present, mostly incompatible with each other. So, what do we have and what do we need?

First, lets take two ingredients, being the basics on parthood from mereology, Ground Mereology (see e.g. (Varzi 2004) for an overview), and requirements motivated by the biomedical ontologies, the parthood relation of the RO. In Ground Mereology, $part\_of$ is reflexive, antisymmetric, and transitive, and proper parthood is asymmetric, irreflexive and transitive. The RO has $part\_of$ twice, once for relating endurants (continuants, like *mary's heart part_of mary's body* at time t) and another $part\_of$ for perdurants (occurrents, processes, like *red hepatization#1 part_of inflammation#1*), and $proper\_part\_of$. For the sake of illustration, we assume for the moment that these ingredients comprise the requirements. The parthood properties were checked against the ontology languages and the results included in Table 1. Thus, currently, no DL-formalised ontology represents parthood as ought to be according to Ontology (mereology), but it is within sight with the draft OWL 1.1 and $\mathcal{DLR}_{\mu}$. Furthermore, and looking at implementations, a peculiarity of DLs is that if the relations are typed differently (distinct domain & range restrictions), such as *either* parthood for endurants *or* relating a part-process to its whole-process, then the relations *must* have different labels to distinguish the two relations. Thus, one $part\_of$ for both types, as described in the RO article (Smith *et al.* 2005) and RO and Cell Cycle online OWL files in the respective 'comment' fields, does not suffice, because a reasoner does not reason over textual comments. However, sticking another label to RO's process-based $part\_of$ in the OWL version can be solved easily.

Biologists' requirements are gradually being met and OWL 1.1 and $\mathcal{DLR}_{\mu}$ already provide opportunities for biomedical ontology developers to incorporate proper parthood relations more comprehensively and consistently throughout the extant ontologies than at present, thereby providing an opening toward achieving better ontology interoperability and data integration.

[4] http://www.opengalen.org/tutorials/crm/tutorial9.html

[5] caCORE v3.2 Technical Guide, 22-12-2006, ftp://ftp1.nci.nih.gov/pub/cacore/caCORE3.2_Tech_Guide.pdf

[6] http://proteinontology.info/documentation.htm

Guidelines for choosing the most suitable formal language

The main question is, of course: what do you want to do with the formal ontology or conceptual model? We discuss some common scenarios in this section, and relate them to several extant bio(medical)-ontologies. First, we have to address computational limitations and under-used features of ontology languages, where the latter provides several opportunities to put up with the former in order to achieve acceptable performance levels when reasoning with very large ontologies and ontologies linked to large data sets. The second part of this section is devoted to the guidelines.

Computational limitations and under-used features

The first step in answering the question is to determine what is more important: getting all details correctly represented, *i.e.*, to represent scientific theories as comprehensive as possible, or automated reasoning support (including query answering) over the ontology or conceptual model. The reason for this either-or choice is the direct proportional relation that exists between the computational complexity of reasoning over an ontology and the expressive power of the language used to formalize the ontology. The computational complexity of a problem indicates the rate at which the resources (*i.e.*, computation time and memory) required to solve the problem grow with respect to the size of the problem's input. For instance, the computational complexity of reasoning in OWL 1.1 is NExpTIME-complete (Cuenca Grau *et al.* 2006) and in the $\mathcal{DLR}$ family is in ExpTIME (Calvanese, De Giacomo, & Lenzerini 1998), whereas the $DL\text{-}Lite$ family remains within polynomial time (Calvanese *et al.* 2006a). Practically, this means that software systems using OWL 1.1 and $\mathcal{DLR}$-formalized ontologies and conceptual models will grow exponentially slower with every increase in the size of the ontology or the amount of data populating the ontology, whereas systems using $DL\text{-}Lite$ will grow only polynomially, as with relational database systems. Hence, the latter can deal with much larger inputs. For instance, ontologies that are populated by more than a few hundred thousand individuals currently may require hours or days when modelled with and queried through expressive languages instead of the desired seconds or minutes, as observed by, *e.g.*, (Marshall *et al.* 2006) with their HistOn ontology about transcription factor binding sites. Classification of protein phosphatases (Wolstencroft, Stevens, & Haarslev 2007) using the ontology was not scalable either. In some cases, the expressivity of a language might render the reasoning problems computationally undecidable (*e.g.*, OWL-Full (McGuinness & van Harmelen 2004)), which means that it is impossible to implement systems which provide automated reasoning support for the full language. These inherent limitations cannot be circumvented by experienced software programmers. This might seem a big problem for adoption of Semantic Web technologies by biology and biomedicine, but is not necessarily so.

Biological and biomedical reality is exceedingly more challenging to represent in an ontology or conceptual model than, say, the enterprise domain, and identifying necessary *and* sufficient conditions (see "Asserted conditions" in Protégé) for DL's 'defined concepts' rarely occurs; *e.g.*, the MGED ontology[7] for microarray experiments, mammalian phenotype[8], BioPax level2[9] for biological pathways, and HistOn have only primitive concepts. Put differently, developing a taxonomy tree-only is already quite an achievement, and *the full expressive power of OWL is not used*. Yet, if one has a 'simple' taxonomy or ontology but still uses a reasoner for expressive ontology languages, it is like trying to kill a mosquito with a sledgehammer: it uses a range of algorithms for descriptions that could be in the ontology, but are not included in the domain ontology. With an ontology that uses a less expressive ontology language, one should be able to take advantage of more efficient reasoning algorithms for the fewer tasks to compute and gain in performance.We illustrate this briefly for several bio-ontologies.

Example. In the same way that OWL 1.1, OWL-DL and OWL-Lite are characterized by a DL, the expressivity used in an ontology build in OWL (or OBO[10] given the OBO-to-OWL mapping[11]) is characterized by a DL which can be identified by analysing the language constructs used in it. We present such an analysis for the previously mentioned ontologies and some other well known bio-ontologies in Table 2 (sample date: 12-2-2007). The table is sorted (approximately) from the most to the least expressive ontologies with respect to the DLs that characterize them (The Description Logics Handbook (Baader *et al.* 2003) has a complete overview of the expressivity of the DLs presented in the table, including further explanations on the different letters; summaries of the letters can be found in, for instance, the 'metrics' option in Protégé and the online complexity navigator[12].). The results were obtained by loading the ontologies into Protégé and SWOOP and using their *DL expressivity metric* facilities to identify the DL that characterized the ontologies. These automated analyses, however, do not take into account important structural information of the ontologies which can further characterize the expressivity used in the ontologies. Put differently, in some cases the expressivity could be made lower by removing or remodelling some redundant or 'odd' assertions (from a logician's point of view).

[7] http://mged.sourceforge.net/ontologies/MGEDontology.php
[8] http://www.informatics.jax.org/searches/MP_form.shtml
[9] http://www.biopax.org/
[10] http://obo.sourceforge.net/main.html
[11] http://www.bioontology.org/wiki/index.php/OboInOwl:Main_Page
[12] http://www.cs.man.ac.uk/ ezolin/logic/complexity.html
[13] http://obo.sourceforge.net/cgi-bin/detail.cgi?nmr
[14] http://obo.sourceforge.net/cgi-bin/detail.cgi?human-dev-anat-abstract
[15] http://www.meteck.org/supplDILS.html
[16] http://obo.sourceforge.net/cgi-bin/detail.cgi?psi-mi
[17] http://obo.sourceforge.net/cgi-bin/detail.cgi?mammalian_phenotype
[18] http://diseaseontology.sourceforge.net/

Ontology	Characterizing DL
ProPreO	$\mathcal{SHOIN}(\mathcal{D})$
BioPAX	$\mathcal{ALCHON}(\mathcal{D})$
Cell Cycle Ontology	$\mathcal{SIN}(\mathcal{D})$
HistOn	$\mathcal{ALCHIF}(\mathcal{D})$
NMR Ontology[13]	$\mathcal{SHF}$
MGED Ontology	$\mathcal{ALEOF}(\mathcal{D})$
Human Developmental Anatomy Ontology[14]	$\mathcal{ALEOF}(\mathcal{D})$
Microbial Loop[15]	$\mathcal{ALCHI}$
Cell Type Ontology	$\mathcal{ALE}(\mathcal{D})$
Gene Ontology	$\mathcal{ALE}(\mathcal{D})$
Protein-Protein Interaction Ontology[16]	$\mathcal{ALE}(\mathcal{D})$
Mammalian Phenotype Ontology[17]	$\mathcal{AL}(\mathcal{D})$
Disease Ontology[18]	$\mathcal{AL}$
FungalWeb[19]	$\mathcal{FL}_0$

Table 2: DL characterization of the expressivity of several bio-ontologies sorted in (approximate) decreasing order with respect to the complexity of the language.

Given the languages and analysis of the examined ontologies, we can see that the current Gene Ontology taxonomies, Protein-Protein Interaction ontology[20], and HistOn, among others, remain within $DL\text{-}Lite_{\mathcal{A}}$ expressivity. The BioPax and MGED ontologies can be adapted easily to match $DL\text{-}Lite_{\mathcal{A}}$ by changing representation of `oneOf` in the ontology (the one-off construct is not supported in $DL\text{-}Lite_{\mathcal{A}}$). In these ontologies, the one-off construct is used to record versioning information through having versioning as instances (strings of text) and subsequently nominalized into a class. This means that versioning is part of the ontology about microarray experiments, whereas it is not necessary to, say, classify versions of an ontology or compute satisfiability. Versioning information certainly needs to be recorded *for* an ontology, but not *in* an ontology; that is, it is information about the ontology, but should not be a component in the ontology-as-logical-theory over which one performs the automated reasoning. By moving this information to comment fields, also Biopax and MGED can take advantage of improved performance in all software implementations compared to their OWL-DL representation. On the other hand, the developers of the Foundational Model of Anatomy and Cell Cycle Ontology want to be as comprehensive as possible, and therefore are served better by OWL 1.1. Subsequently, one can extract a 'light' version of such comprehensive ontologies into $DL\text{-}Lite$ to aid implementation of, *e.g.*, database integration in the biomedical domain. $\diamondsuit$

Having illustrated usage of ontology languages and having provided an example of adjusting an ontology to a language of much lower complexity that will improve performance, we now can proceed to the guidelines.

[19] http://www.cs.concordia.ca/FungalWeb/

[20] http://psidev.sourceforge.net/mi/xml/doc/user/index.html

Ontology language choices

Based on the preceding analysis of language features, computational limitations, and (under-)usage of language features, we propose several guidelines to choose the (relatively) optimal ontology language for the core intended tasks.

I. Comprehensiveness

a. *No computation*. The user can choose freely the language that covers to the best extent the expressive requirements of the ontology. Suitable languages are the OWL 1.1 and $\mathcal{DLR}$ families, or to resort to other logics that are currently largely outside of the scope of the Semantic Web, such as first- or higher order logics, temporal logics, epistemic logic etc.. For instance, to represent a scientific theory as comprehensive as possible and for foundational ontologies, such as BFO[21] and GFO (Herre & Heller 2006).

b. *Some computation desired and plenty of time and memory is available*. A decidable language has to be used that should have the lowest computational complexity as possible while covering the requirements for expressiveness. The size of the ontology or the data will be limited by the resources at hand, that is: either a large ontology of universals or a small one that can be linked to a small amount of data. Languages suitable for this setting are OWL-DL, OWL 1.1, the $\mathcal{DLR}$ family. For instance, constructing integrated conceptual models that are used 'off-line' for eventual data integration and developing reference ontologies, such as the FMA, for particular subject domains.

II. Computation

a. *Computing time and memory are an important component*. This is a grey area as to what constitutes a reasonable amount of waiting time, and either OWL-DL, OWL-Lite or $DL\text{-}Lite$ could be used: the former two if there is relatively little data (with as rule-of-thumb, certainly less than hundred thousand instances)

[21] http://www.ifomis.uni-saarland.de/bfo/home.php

and if the ontology is small (less than a few hundred DL-concepts); $DL\text{-}Lite$ can be used in all scenarios. For instance, classification of defined DL-concepts and instance classification, such as with aforementioned protein phosphatases, and online support for (semi-)automated database integration using conceptual models or an ontology.

b. *Computing time and memory are critical*. The user has less expressive power at disposal, limiting the accuracy of the ontology compared to item I, but its size and the amount of data linked to the ontology can be as for relational databases. Languages suitable for this setting are those in the $DL\text{-}Lite$ family. For instance, to pose complex queries over the data, like microarray data and data about large genomes, through ontologies such as the GO, MGED ontology, and HistOn.

With these main four distinctions, one could construct a decision procedure, as in "if you want an ontology to do x, then...". However, the four distinctions remain and can be reused for any new scenario, whereas a decision tree would have to be updated upon each usage variation.

One can argue that ontologies ought to be purpose-independent to obtain maximum reuse for both representing scientific theories and a wide range of Semantic Web applications. However, one cannot have it both ways at the same time. A nearby solution is to let the computer program automatically 'simplify' an ontology when users demand performance over expressivity. First steps in this direction have been taken already: mapping $DL\text{-}Lite_{\mathcal{A}}$ to OWL 1.1 (by one of the authors, MR), and an early prototype of the QuOnto tool (Acciarri *et al.* 2005) is available online[22], which is the first system that can answer complex queries (union of conjunctive queries) expressed over ontologies whilst capable of managing millions of instances. Further, there is an implemented transformation from $\mathcal{DLR}$ to the DIG API in the iCOM tool[23] (Franconi & Ng 2000), so that current automated reasoners, such as Pellet and RACER, can be used to reason over formal conceptual models that are the abstract representations of the physical databases. These are research-grade tools, however, that focus on verifying that the theory is implementable and much less on user-friendliness and usability for domain experts. They could be useful for bio-ontologists and bioinformaticians researchers to examine the latest theoretical results for addressing the bio-sciences Semantic Web requirements of linking data to the ontologies and using this combination efficiently.

Conclusions

Based on, and motivated by, a comparative assessment of ontology- and formal conceptual modelling languages, current bio-ontologies and their usage, and prospective scenarios for ontology-based and ontology-mediated tasks, we provided suggestions for choosing the optimal ontology language for the task. Although it is expected that ontology languages develop further, the main trade-off between expressivity and usability in data-intensive biomedical information systems remains. Regarding the currently supported modelling features in the considered languages, this results in having to choose between, e.g., proper representation of parthood relations versus n-ary relations and qualified number restrictions versus role values.

Current research comprises mapping $DL\text{-}Lite$ to OWL 1.1 and incorporating a DIG API for the QuOnto system, which will enable easy adoption of the $DL\text{-}Lite$ languages by current OWL/Protégé users. We plan to conduct a more comprehensive analysis of the (under-)used ontology language capabilities, develop algorithms for 'lite'-izing expressive ontologies to use for ontology-based data access and integration, and reasoning services for bio-ontologies to better match the 'offers-and-demands'.

Acknowledgments. The authors thank Diego Calvanese for helpful suggestions on an earlier draft.

[22] http://www.dis.uniroma1.it/∼quonto.

[23] http://www.inf.unibz.it/ franconi/icom/.

References

Acciarri, A.; Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Palmieri, M.; and Rosati, R. 2005. QUONTO: QUerying ONTOlogies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, 1670–1671.

Antezana, E.; Tsiporkova, E.; Mironov, V.; and Kuiper, M. 2006. A cell-cycle knowledge integration framework. In *Data Integration in the Life Sciences (DILS2006)*, volume 4075 of *LNBI*, 19–39. Springer Verlag.

Artale, A., and Franconi, E. 1998. A temporal description logic for reasoning about actions and plans. *J. of Artificial Intelligence Research* 9:463–506.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press.

Bandini, S., and Mosca, A. 2006. Mereological knowledge representation for the chemical formulation. In *2nd Workshop on Formal Ontologies Meets Industry 2006 (FOMI2006)*, 55–69.

Bechhofer, S.; Horrocks, I.; and Turi, D. 2005. The OWL Instance Store: System description. In *Proceedings of CADE-20*, LNCS. Springer Verlag.

Berardi, D.; Calvanese, D.; and De Giacomo, G. 2005. Reasoning on UML class diagrams. *Artificial Intelligence* 168(1–2):70–118.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. DL-Lite: Tractable description logics for ontologies. In *Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005)*, 602–607.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2006a. Linking data to ontologies: The description logic *dl-lite*$_{\mathrm{a}}$. In *Proc. of the 2nd Workshop on OWL: Experiences and Directions (OWLED 2006)*.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2006b. Data complexity of query answering in description logics. *Proceedings of KR-2006* 260–270.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1998. On the decidability of query containment under constraints. In *Proc. of the 17th ACM Symp. on Principles of Database Systems (PODS'98)*, 149–158.

Calvanese, D.; De Giacomo, G.; and Lenzerini, M. 1999. Reasoning in expressive description logics with fixpoints based on automata on infinite trees. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, 84–89.

Cuenca Grau, B.; Horrocks, I.; Parsia, B.; Patel-Schneider, P.; and Sattler, U. 2006. Next steps for OWL. In *Proc. of the Second OWL: Experiences and Directions Workshop (OWLED-2006)*.

Franconi, E., and Ng, G. 2000. The i.com tool for intelligent conceptual modeling. In *Proc. of the 7th Int. Workshop on Knowledge Representation meets Databases (KRDB 2000)*, 45–53. CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-29/.

Gene Ontology Consortium, . 2004. The Gene Ontology GO database and informatics resource. *Nucleic Acids Research* 32(1):258–61.

Herre, H., and Heller, B. 2006. Semantic foundations of medical information systems based on top-level ontologies. *Knowledge-Based Systems* 19:107–115.

Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The even more irresistible $\mathcal{SROIQ}$. *Proc. of KR-2006* 452–457.

Kazic, T. 2006. Putting semantics into the semantic web: How well can it capture biology? *Proc. of Pacific Symposium in Biocomputing* 11:140–151.

Keet, C. M. 2007. Mapping the Object-Role Modeling language ORM2 into Description Logic language $\mathcal{DLR}$ifd. Technical Report KRDB07-2, Faculty of Computer Science, Free University of Bozen-Bolzano, Italy. arXiv:cs.LO/0702089v1.

Marshall, M. S.; Post, L.; Roos, M.; and Breit, T. M. 2006. Using semantic web tools to integrate experimental measurement data on our own terms. In *International Workshop on Knowledge Systems in Bioinformatics (KSinBIT'06)*, volume 4277 of *LNCS*, 679–688. Springer Verlag.

McGuinness, D. L., and van Harmelen, F. 2004. OWL Web Ontology Language Overview. W3C Recommendation. http://www.w3.org/TR/owl-features/.

Rosse, C., and Mejino Jr, J. V. 2003. A reference ontology for biomedical informatics: the foundational model of anatomy. *J. of Biomedical Informatics* 36(6):478–500.

Ruttenberg, A.; Rees, J.; and Zucker, J. 2006. What biopax communicates and how to extend owl to help it. In *Proc. of the Second OWL: Experiences and Directions Workshop (OWLED-2006)*.

Schulz, S., and Hahn, U. 2004. Parthood as spatial inclusion–evidence from biomedical conceptualizations. *Proc. of KR-2004* 55–63.

Schulz, S.; Hahn, U.; and Romacker, M. 2000. Modeling anatomical spatial relations with description logics. *AMIA 2000 Annual Symposium* 779–783.

Seshadri, R.; Kravitz, S.; Smarr, L.; Gilna, P.; and Frazier, M. 2007. CAMERA: A community resource for metagenomics. *PLoS Biology* 5(3):e75.

Smith, B.; Ceusters, W.; Klagges, B.; Köhler, J.; Kumar, A.; Lomax, J.; Mungall, C.; Neuhaus, F.; Rector, A.; and Rosse, C. 2005. Relations in biomedical ontologies. *Genome Biology* 6:R46.

Smith, B.; Kusnierczyk, W.; Schober, D.; and Ceusters, W. 2006. Towards a reference terminology for ontology research and development in the biomedical domain. In *International workshop Biomedical Ontology in Action (KR-MED 2006)*.

Varzi, A. C. 2004. Mereology. In Zalta, E. N., ed., *Stanford Encyclopedia of Philosophy*. Stanford, fall 2004 edition. http://plato.stanford.edu/archives/fall2004/entries/mereology/.

Wolstencroft, K.; Stevens, R.; and Haarslev, V. 2007. Applying owl reasoning to genomic data. In Baker, C., and Cheung, H., eds., *Semantic Web: revolutionizing knowledge discovery in the life sciences*. Springer: New York. 225–248.