Dependencies between Ontology Design Parameters

C. Maria Keet

KRDB Research Centre,
Free University of Bozen-Bolzano,
Piazza Domenicani 3
Bolzano, Italy
Fax: +39 0471 016009
E-mail: keet@inf.unibz.it

Abstract: Development and use of ontologies is increasing, but hampered by new challenges, such as determining which ontologies to reuse and which language to use. Ontology development methodologies provide scenarios, but they do not address the dependencies between the permutations at the different stages in the development process. To improve the efficiency and effectiveness of this endeavour, we group inputs into parameters and examine the dependencies between them. The parameters considered are: nine main purpose(s) of the ontology, reuse of the three main types of ontologies, eight different ways for bottom-up ontology development, eight ontology languages, and four types of reasoning services. All dependencies between these parameters have been assessed and useful combinations are motivated. The dependencies are due to, primarily, computational challenges and types of subject domain of the ontologies. This analysis is assessed against a set of ontologies and a survey among ontology developers, whose results concur with the theoretical assessment.

Keywords: ontologies; ontology development: bottom-up ontology development; ontology reuse; ontology language; ontology development methodology; OWL.

Reference to this paper should be made as follows: C. M. Keet. (201x) 'Dependencies between Ontology Design Parameters', *Int. J. IJMSO*, Vol. 1, No. x, pp. xx–xx.

Biographical notes: Maria Keet is a non-tenured Assistant Professor with the KRDB Research Centre at the Free University of Bozen-Bolzano, Italy, where she also received her PhD in Computer Science in 2008. Her research interests include logic-based knowledge representation and ontology theory and applied to biological data, conceptual data modelling, and granularity, which has resulted in more than 30 peer-reviewed publications. She has served as PC member for several conferences and reviewed for international journals. She also holds and MSc specialised in microbiology form Wageningen University and an MA in Peace & Development studies from the University of Limerick. Before returning fulltime to academia in 2002, Maria has worked for 3.5 years as systems engineer in the IT industry.

1 Introduction

A rapid increase of development and publication of new ontologies has occurred in recent years, reaching well over 200 ontologies and an increasing number is under development. Earlier modelling issues are currently being, or already have been, addressed or even surpass the required solution up to a point that is has generated new problems. Notable advances are the foundational ontologies, the mushrooming of freely available bio-ontologies, and the W3C standard for OWL as common ontology language in the Semantic Web. However, solving one problem moves the goal-posts. For instance, which ontologies are reusable for one's own ontology, what are the consequences choosing one over the other? The successor of OWL, OWL 2 (Motik et al., 2009b), actually has 5 languages tailored for different purposes: which one should be used for what and when? Ontology development methodologies, of which one might want to assume to hold the answers, offer one way for how one can develop an ontology, or present (subsets of) possible scenarios, such as 'Building an ontology by reusing ontologies or ontology modules', which do not say which type of ontologies there are and which ones are amenable for reuse or may even be intended for reuse, what the purpose of the gain of such reuse is, and which version of that ontology is most suitable for reuse within the chosen purpose of the new ontology. Put differently, there is a difference between using, say, score cards that one has to fill in (e.g., in Suarez-Figueroa et al., 2008) and to have an overview of the available options and the consequences for choosing one or the other, i.e., the *interaction* between the options at different stages in the methodology. While there exists literature that provide incomplete and overlapping options at some of the stages in a methodology, first and foremost they lack the dimension of the interactions and consequences. In addition, despite that several ontology development methodologies are available, many developers actually do not use them[1]. Which strategies and additional methods do developers employ when they do not adhere to a particular methodology? What are the possible permutations at each step? What are the choices they face, and can they be categorised in types of choices? Does a choice for one method, tool, language and so forth lock one into another, or are some of them even mutually exclusive? If so, why do they go together or are they mutually exclusive, and can or should that be changed?

In this paper, we seek to answer these questions by, first, integrating a literature analysis, our own experiences in ontology development, and results of a survey we conducted so as to identify the key parameters. The ontology design parameters are structured into five main groups in order to provide a succinct and clear overview of the principal options at several different stages of the ontology development process. The main five groups are: purpose, reuse of ontologies, bottom-up ontology development strategies, the language to represent the ontology, and the reasoning services. The second part presents the novel contribution with the outcome of the analysis regarding the practical and theoretical dependencies between the key parameters; this is summarised in Table 1. It addresses interdependencies such as “given a purpose which ontology language suits best?” and the interactions between purpose, language, and bottom-up development strategy. The dependencies themselves have to do primarily with inherent computational challenges, insufficient tools and limitations of existing ones, and usage of domain ontologies. We also present several useful combinations of the five parameters, which can guide the novice developer toward comparatively suitable and easy combinations. In addition to the theoretical analysis, we also consider the background and setting of a set of randomly selected ontologies and those that the survey respondents were involved in, which indeed do tend to adhere to the proposed dependencies between the parameters. The results presented in this paper can be used in, e.g., the design of a software-supported selection procedure as part of a flexible ontology development methodology, so that ontology development will not only be made more easily accessible concerning where to commence given one’s resources and desired goals but also shorten the development process by proposing combinations of parameters that lead to a feasible implementation scenario.

The remainder of the paper is structured as follows. After clarifying the notions of ontologies and methods in Section 2, we proceed to list the design parameters and illustrate them insofar as they bear an effect on the dependencies (Section 3). The dependencies between the parameters are described and discussed in Sections 4 and 5, and we close with conclusions in Section 6.

2 Preliminaries

To put the main section in the right context, we first have to address some terminological aspects about what is—or can be—meant to be an ontology, and the notions of methods and methodologies themselves.

2.1 Ontologies and ontological commitments

The most quoted definition of an ontology as a specification of a conceptualization by Gruber (1993) has been discussed widely and refined over the years. For instance, a more detailed definition is that it is “a logical theory accounting for the *intended meaning* of a formal vocabulary, i.e. its *ontological commitment* to a particular *conceptualization* of the world. The intended models of a logical language using such a vocabulary are constrained by its ontological commitment. An ontology indirectly reflects this commitment (and the underlying conceptualization) by approximating these intended models.” (Guarino, 1998), or a formal representation of reality instead of a representation of our conceptualisation of entities in reality. Practically from an engineering perspective, the reality-vs.-conceptualization representation is a philosophical quarrel which acts out as differences of opinion during the modelling stage when there is a disagreement about an axiom if it has to represent the way how some vocal or large groups of people think about something or if it also has to be the most truthlike with respect to the reality it aims to represent. For instance, if $Whale$ should be represented as a subclass of $Fish$ or of $Mammal$: the former is a biologically incorrect assumption by many people and may end up as such in a folksonomy, but, in accordance with the reality representation and the notion of what it is to be a mammal, only the latter should be represented in an ontology.

Where ontological commitments affect not only the subject domain but also the language, are considerations such as an ontology of universals (or concepts or classes), of individuals, or if it can be a combination of the two. From the philosophers’ and modellers’ side, one separates an ontology of universals from one of particulars. However, the main authors behind the W3C standardised ontology language OWL simplified the definition of an ontology into it “being equivalent to a Description Logic knowledge base” (Horrocks et al., 2003), hence, it being a combination of the intensional terminological axioms (TBox) *and* the extensional assertional axioms about instances (ABox). In the remainder of the paper we assume a clear separation between the classes and instances and, given that most existing ontologies are in the area of the healthcare and life sciences that focus on type-level generalisations, a logical theory in the sense of TBox statements only.

2.2 Methods and methodologies for ontology development

Methodologies seek to provide a structured way of developing ontologies from start to deployment to maintenance. This paper analyses steps and methods that do, or can, form part of ontology development methodologies so as to augment the methodologies with the dimension of the interdependencies between the steps and methods they contain. Examples of decision steps are which foundational ontology to reuse, if any, and the choice of language. Examples of methods that contribute to methodologies for ontology development are, among others, Ontology Design Patterns that offer small, reusable, pieces of an ontology or modelling pattern of usage (Presutti et al., 2008), OntoClean to design an ontologically well-formed taxonomy (Guarino and Welty, 2004), 'debugging' ontologies to pinpoint to errors (Horridge et al., 2008), and NLP-based ontology learning (Witte et al., 2007) to speed up finding candidate terms and relations for one's domain ontology. Methodologies can be grouped according to different axes, such as aimed at single, one location ontology development vs. distributed ontology development, or paper-based vs. tooling support and have varying levels of support for managerial tasks such as project planning and versioning; for instance, METHONTOLOGY, On-To-Knowledge, KACTUS, MoKi, and NeOn (Fernandez et al., 1999; Suarez-Figueroa et al., 2008; Ghidini et al., 2009) (the reader is referred to (Corcho et al., 2003) for an analysis of requirements and a comparison between such ontology development methodologies). These methodologies take a waterfall or iterative approach without revealing dependencies between their different stages. However, such assumed independence neither holds in practice—be it with respect to the soft skills or the available tools—nor is it feasible even theoretically, as we shall see in Section 4.

3 Design parameters

In this section we describe the five principal categories of design parameters: purpose of the ontology, reusing ontologies, bottom-up development of ontologies, representation languages, and reasoning services. For each one, we list the parameters, and elaborate on them afterwards insofar as it affects, or will contribute to, the explanation of the dependencies between them.

3.1 Purposes of the ontologies

One can take into account the possible aims for which the ontology will be developed. However, this is anathema for the ontology purist, because an ontology is supposed to be implementation independent—even irrespective if an application will be linked to it or will have any computational use at all—and as such, an ontology has the sole purpose of representing reality, our conceptualisation of reality, or some other domain of interest. A caveat with choosing explicitly for a specific purpose, is that a few years after initial development of the ontology, it may get its own life and be used for other purposes. This can require a re-engineering of the ontology (currently being carried out with the Gene Ontology (GO) and Foundational Model of Anatomy (FMA)), which generally costs more resources than anticipating upfront that an ontology may be used for different purposes. Nevertheless, in the practice of ontology engineering, purposes tend to play an important role and oftentimes are even expected to be formulated at the commencement of ontology development. Based on a literature review and survey we carried out (Alberts et al., 2008), the different types of purposes can be summarised as follows (each with a reference that provides a typical example):

1. Querying data by means of an ontology (ontology-based data access) through linking databases to an ontology (Poggi et al., 2008; Alberts et al., 2008; Rodriguez-Muro et al., 2008);

2. Data(base) integration, most notably the strand of applications initiated by the Gene Ontology Consortium and a successor, the OBO Foundry (Gene Ontology Consortium, 2004; Smith et al., 2007);

3. Structured controlled vocabulary to link data(base) records and navigate across databases on the Internet, also known as 'linked data' (Bizer et al., 2009, in press);

4. Using it as part of scientific discourse and advancing research at a faster pace (Keet, 2005; Madin et al., 2008), including experimental ontologies in a scientific discipline and usage in computing and engineering to build prototype software;

5. Coordination among and integration of Web Services (Goble et al., 2007);

6. Incorporating the ontology in an ontology-driven information system destined for run-time usage, such as in scientific workflows, multi-agent systems, ontology-mediated data clustering, and user interaction in e-learning (Henze et al., 2004; Kim et al., 2006; Zhou et al., 2005);

7. Ontologies for natural language processing, including development and use of ontologies in applications such as annotating and querying Digital Libraries and scientific literature, Question-Answering systems, and materials for e-learning (Alexopoulou et al., 2008; Vila and Ferrández, 2009; Witte et al., 2007);

8. As full-fledged discipline "Ontology (Science)" (Smith, 2008), where an ontology is a formal, logic-based, representation of a scientific theory;

9. Tutorial ontologies to learn modelling in the ontology development environment (e.g., the wine and pizza ontologies (Horridge et al., 2004)).

Practical examples have their specific peculiarities and implementation, but the *type of purpose* will fit in one of the above, although e.g., Denaux et al. (2009)'s experiment to extract purpose phrases from ontology documentation of geography ontologies might be extended to a larger corpus of ontologies that have better ontology annotations and thereby contribute to a more refined categorisation than the list presented here.

Observing the purposes, it is noteworthy that five items concern data integration in some way, being items 1, 2, 3, 5, and 6. Recollecting the data integration strategies that have been developed over the past 20 years, they are: (i) Physical schema mappings with Global As View (GAV), Local As View (LAV), or GLAV, (ii) Conceptual model-based data integration, (iii) Data warehouses, (iv) Data federation, (v) Data marts, (vi) Services-mediated integration, (vii) Peer-to-peer data integration, and (viii) Ontology-based data integration, being on top of, at least, i, ii, (possibly also iii iv, v, vi) through an ontology or linked data by means of an ontology. While some of them can be realised without ontologies, ontologies have proven their value in the endeavours. For instance, the GO for linking data (Gene Ontology Consortium, 2000), connecting different legacy tools (Athanasiadis et al., 2009), data federation (Amoroso et al., 2008), RDF-ized warehouse with an ontology (Belleau et al., 2008) as well as first steps toward RDF-focussed database federation (Cheung et al., 2009). It is important to observe that for such 'ontologies for data integration', in fact, an ontology serves different purposes within the integration scenario and thereby brings forth different requirements for the ontology language, tools, and ontological commitments.

A more challenging purpose that entails at least data linking, if not full integration, is using ontologies in scientific workflows, such as Wings (Kim et al., 2006) that uses both the ontology and its reasoning services for the generation of workflows, keeping consistency, and linking sections of a workflow, and the Taverna workbench (Goble et al., 2007), which uses a set of ontologies to manage the data and information flow by means of a domain ontology, workflow ontology, and services ontology. The latter, in turn, puts demands on interoperability not only among subject domain ontologies, like (Smith et al., 2007), but also among ontologies with orthogonal content and complementary purposes, so that it requires a careful orchestration.

3.2 Reusing ontologies

With the uptake of ontology development and mushrooming of ontologies made available on the Web, ontology repositories and semantic search systems, such as Swoogle[2], the TONES Ontology Repository[3], and BioPortal[4] can be helpful. However, not all ontologies are just more of the same and equally suitable for reuse in one's own ontology. The main types of ontologies that potentially can be reused in part or whole are:

1. Foundational ontologies that provide generic top-level categorisations;
2. 'Reference ontologies' that contain the main concepts of a subject domain;
3. Domain ontologies that have a (partial) overlap with the new ontology;
4. For each of items 1-3, resource usage considerations, such as
    (a) The availability of the resource, such as openly available, copyright, and usage restrictions;
    (b) If the source is being maintained or an abandoned one-off effort;
    (c) The ontology is a result of a community effort, research group, or if it has already some adoption or usage;
    (d) If it is subject to standardization policies or has stable releases;
    (e) If the ontology is available in the desired or required ontology language.

The foundational ontologies can give a head-start in the development by providing a basic structure, such as endurants being disjoint from perdurants, types of processes, attributes (qualities), and a set of basic relations; e.g., GFO, DOLCE, BFO, RO (Herre and Heller, 2006; Masolo et al., 2003; Smith et al., 2005). The use of such ontologies (or not) requires one to decide how to represent certain kinds of knowledge, most notably 'attributes'. In the traditional sense, an attribute $a \in A$ is a binary functional relation from an object o in a universe of objects U to a value v in the value domain V (i.e., $A \mapsto U \times V$), for instance, a person's age as $hasAge \mapsto Person \times Age$ or a flower's colour $hasColour \mapsto Flower \times Colour$. None of the foundational ontologies use this approach to represent properties of objects (Borgo and Masolo, 2009). Instead, properties such as *Colour* are entities (unary predicates) in their own right and are related simply with a binary relation to objects; e.g. in BFO, *Colour* is a kind of dependent continuant that *inheres_in* an independent continuant (*Flower*) and in DOLCE, such 'attributes' are subsumed by the category *Quality* that relates to *Endurant* (roughly: objects) or *Perdurant* (informally: processes) by means of the *qt* relation and each quality has a separate relation to *qualia*, which can be, among others, *Temporal Region* or *Spatial Region*. That is, the compact notion of 'attribute' in information systems and conceptual data modeling languages, such as ER and UML class diagrams, is flattened out in the foundational ontologies into at least *two* relations and *three* entities (unary predicates). From an ontological perspective, the verbosity of this attribute-free approach is an advantage with respect to modelling 'attributes' more precisely (but it may not necessarily be from an implementation perspective) and it has been shown to greatly enhance transparency and reusability[5], which is a sought-after characteristic for foundational ontologies.

Reference ontologies are more restricted in scope, but they are also intended for reuse, such as an ontology of measurements, of time (Hobbs and Pan, 2004) and 'top-level' ontologies for a domain, such as BioTop[6]. Domain ontologies, in turn, can build upon such foundations and expand on it for the particular subject domain at hand, such as for traits of rice in Gramene that extends GO and marine microbial loops that reuses DOLCE (Jaiswal et al., 2002; Keet, 2005). Generally, then, the developer adheres to the foundational ontology commitments, but does not debate and decide on it anymore.

Further down the line are so-called 'application ontologies', which, however, are another term for logic-based conceptual data models. These type of ontologies are explicitly *not* included in the list above, because they will generate more problems than they solve: in computing and engineering, ontologies themselves were proposed to address computationally the commonalities and mismatches in conceptual data models, in the sense of providing an *application-independent* representation at a higher abstraction level (see also Jarrar et al. (2003); Schreiber (2007) on this topic). For instance, in a conceptual data model for a zoo, one my want to have a class *Llama* and a key or reference scheme in numbers, whereas another database management system may have the identification of llamas represented with a string; then it is through the common understanding and precise representation of what a llama is—represented in the ontology—that an agreement can be established that both software systems deal with representations of llamas. If one designates such conceptual data models to be ontologies, then one cannot properly use ontologies anymore for data integration and thereby deprive oneself of a powerful solution and generate new problems by (i) taking away one layer in the successive abstraction steps and (ii) inventing and proliferating ontology mapping languages (Brockmans et al., 2009) and supporting tools.

The fourth item has nothing to do with the content of the ontology, but are orthogonal considerations that interfere with the choice which ontology to reuse for one's purpose. Aspects (a-d) are 'soft' considerations: if an ontology has usage restrictions, is an abandoned one-off effort, and has not been adopted by other ontologists, then one probably should think twice before using it, but there nevertheless can be compelling reasons to do so. Item (e) reveals a dependency with other design parameters, and will be discussed in Section 4.

3.3 Bottom-up development of ontologies through ontology learning

Although ontology developers will find something of use in the available ontologies, they often have to develop part of the ontology themselves. There are several strategies to speed up this labour- and knowledge-intensive task, which that focuses on extracting the subject domain semantics present in 'legacy' (i.e., currently operational) sources. The type of sources and principal techniques (ordered in decreasing order of formality) are:

1. Reuse of other knowledge-based representations: conceptual data models of database and application software, such as UML diagrams, ER diagrams, and ORM models;

2. Extraction of types from a database physical schema and data in databases (i.e., database reverse engineering) and object-oriented software applications, and least common subsumer and clustering to infer new concepts;

3. Abstractions from or formalisations of models in textbooks and diagram-based software;

4. Thesauri and other structured vocabularies;

5. Other (semi-)structured data, such as spreadsheets and company product catalogs;

6. Text mining of documents to find candidate terms for concepts and relations;

7. Terminologies, lexicons, and glossaries;

8. Wisdom of the crowds tagging, tagging games, and folksonomies;

This list is roughly in concordance with Suarez-Figueroa et al.'s (2008) analysis, who also discuss the various typologies of non-ontological resources for bottom-up development of ontologies. In the remainder of this section, characteristics of the source material and approaches will be highlighted insofar as they interfere with respect to the dependencies between the different ontology design parameters.

The first two items tend to be grouped together, but are quite distinct. Reuse of conceptual data models seems to require only a mapping or transformation (i.e., approximation) to a suitable ontology language. However, UML, ER, and ORM do not exactly fit into an OWL species or even one unique Description Logic (DL) language (Artale et al., 2007a; Berardi et al., 2005; Keet, 2007), for the mere facts that ORM is undecidable and there is no single formalisation of UML class diagrams or ER. Practically, this means that one has to develop transformations for each ontology language and each tool that offers conceptual modelling functionalities. In addition, it makes a certain commitment on how to represent 'attributes' (cf. Section 3.2): transformations from UML and ER will be heavy in the use of OWL's data properties, whereas ORM's philosophy is more in line with the foundational ontologies' approach to flatten them into multiple relations.

The sources in item 2 are further away from an ontology than those of item 1. For the sake of simplicity, let us ignore issues such as data duplication, violations of integrity constraints, and denormalised databases. Data stored in the database, which are mathematically instances, sometimes are thought of by the domain expert as representing concepts; e.g., the terms in the GO are intended to denote concepts (universals), but in the GO database they are stored as values in a 2-column table. Biological databases as well as databases at the back-end of content management systems have a mixture of the two so that some database data have to be imported into the ontology as concepts (OWL classes), whereas other data in the database have to be imported into the OWL ABox and therewith transformed from values into objects. This is a well-known reverse engineering challenge, even for reverse engineering to a conceptual data model (Hainaut et al., 1993), let alone if one wants to take the even larger jump directly to an ontology (Lubyte and Tessaris, 2007).

Bottom-up development of ontologies by using other diagrams and its drawing tools is relatively under-explored. Software for drawing diagrams in, predominantly, the life sciences, such as PathwayAssist[7] to visualise metabolic pathways, essentially have their own icon vocabulary and diagrams. This informal but structured representation of information can be exploited to develop automatically a preliminary version of a domain ontology by (i), formalizing the 'icon vocabulary' in a suitable logic language, (ii) choosing a foundational ontology (taxonomy, relations), (iii), categorising the formalised icons accordingly, (iv) loading each diagram into the ontology, and (v) verifying the results with the domain expert, which has been experimented with using STELLA models for ecology and environmental sciences (Keet, 2005).

There are many terminologies and thesauri and a few of them are being adapted for the Semantic Web, such as the NCI Thesaurus[8]. These endeavours tend to end up as bottom-up development instead of a straightforward reuse of the resource and formalisation into SKOS or OWL (Soergel et al., 2004). The reasons for this is that a thesaurus' core relations are broader term (BT), narrower term (NT), and related term (RT), and some auxiliary ones (e.g., UF/USE), which are semantically heavily overloaded relations: BT/NT is not the same as *is_a* and RT can be any type of relation, and, consequently, those relationships are used inconsistently. In addition to this low ontological precision, a categorisation of basic types such as those of foundational ontologies is absent and many terms and relations are actually a lexicalisation of a conceptualisation, so that one has to manually reassess the whole thesaurus again in order to add the implicit semantics explicitly. Thus, simply formalising a thesaurus does not suffice to obtain an ontology. An example of elaborate 'ontologising' thesauri in the fisheries domain, which were augmented with a DOLCE foundation, is described in (Gangemi, 2005).

Another approach that initially seemed an easy route to bottom-up development, is the use of Natural Language Processing (NLP) techniques. The activities can be divided into development of the TBox by means of searching for candidate terms and relations in text documents, and ontology populations (ABox) that amounts to document retrieval enhanced by ontologies and tagged text (e.g., GoPubMed (Dietze et al., 2008)). The former is known to be challenging and an iterative process that requires considerable domain expert input (Alexopoulou et al., 2008; Gliozzo et al., 2007). The main reasons for the problem of not finding immediate matches are the ambiguity of natural language and the choice to collapse the notion of ontology with natural language instead of using them as orthogonal or complementary components for which a lexicalised ontology is more suitable (Cimiano et al., 2007).

Last, one can try to squeeze out the little semantics available in spreadsheets, using perhaps RDF as an intermediate representation (Han et al., 2006) (but see also Madin et al. (2008)). If also this fails to extract useful terms and relations, one could resort to the 'wisdom of the crowds', but this also depends on the subject domain so that at times 'the crowd' are students in the discipline, and enhanced by a gaming component. An alternative to this approach is to let the reasoner compute a concept description based on sample instances (Baader et al., 2007).

3.4 Representation languages

Depending on the purpose(s) and, in practice, available resources, such as time, money, domain experts, and available baseline material, one tends to end up with either (a) a very large simple ontology, i.e., mostly a taxonomy without, or with very few, properties (relations) linked to the concepts, where 'large' is, roughly, $>$ 10000 concepts, so that a simple representation language suffices; (b) a large and elaborate ontology, which includes rich usage of properties, defined concepts, and, roughly, requiring the language expressiveness of that of expressive DL languages, such as OWL-DL; or (c) a small and very complex ontology, where 'small' is, roughly, $<$ 250 concepts, and requiring a very expressive ontology language. Put differently, a separate dimension that interferes with the previous parameters, is the representation language, because some languages scale up better computationally than others. Moreover, certain choices for reusing ontologies or legacy material, or goal, may lock one into the language that will be used to represent the ontology. To obtain a better view on this aspect, we address older and newer ontology languages, and pointers to various extensions.

3.4.1 Older Knowledge Representation and Ontology Languages

With "older" languages, we refer to the pre-OWL stage, where ontologies were developed in a logical language of one's preference. The languages of choice from the side of AI were, among others, first order predicate logic, knowledge interchange format (KIF), KL-ONE, Loom, conceptual graphs, DL, and F-logic, and later also DAML and OIL and DAML+OIL. From a domain expert side, some variations, extensions, or simplifications have been made to these languages and a subsequent renaming has occured. For instance, the OBO format of the GO and similar ontologies is a directed acyclic graph with additional house-keeping items, the Protégé development tool had originally a frames-based system, and TAMBIS's and GALEN's GRAIL language is based on a DL. Then, even if ontologies adhered to the same language, say, full first-order logic, then the computational representation in software still can be different, such as the syntax of the Prover9 and Isabelle theorem provers. Each of these languages have their advantages and disadvantages, but what is problematic in general is that there are so many languages that are not always fully interchangeable and that modelling workarounds to mitigate a language's limitations are often poorly documented, if at all. This complicates ontology reuse and integration, development, use of tools for bottom-up development of ontologies, and ontology-driven information systems. Hence, a standardized ontology language is a step forward to solve ontology interoperation challenges.

3.4.2 Web Ontology Languages OWL and OWL 2

By 2004, things had converged to the W3C standardised Web Ontology Language OWL (Horrocks et al., 2003; McGuinness and van Harmelen, 2004), which is useful from at least the domain modeler's and tool developer's perspective to have one common format to represent an ontology. In addition to choosing OWL *a priori*, core ontology developers as well as user-bases of some of the older ontology languages have chosen to provide converters from the prior language to an OWL language, such as oboinOwl (Golbreich and Horrocks, 2007), FMA to OWL (Zhang et al., 2006), and both DOLCE and BFO are now available also in one or more OWL versions.

From a formal and Semantic Web perspective, OWL is placed on top of RDF(S) and has its basis in Description Logics (DL) languages and reasoning services (Horrocks et al., 2003; Baader et al., 2003). OWL actually consists of three languages of which two are DL-based species, OWL-DL and OWL-Lite, and an OWL full, which is an undecidable RDFS species. Its successor OWL 2 consists of a full species that is again based on RDFS, and four DL-based species: OWL 2 DL and three 'lighter' versions, being OWL 2 EL, OWL 2 QL, and OWL 2 RL (Grau et al., 2008; Motik et al., 2009b,a). The main motivation for including four DL-based ontology languages in the standard, is to allow tailoring the choice of ontology language to fit best with the purpose of the ontology, and in particular for a *scalable* and *multi-purpose* Semantic Web. OWL 2 DL is the most expressive one and based on the DL language $\mathcal{SROIQ}$ (Horrocks et al., 2006), whereas OWL 2 EL and OWL 2 QL are computationally well-behaved fragments to achieve better performance with larger ontologies and ontologies linked to large amounts of data in secondary storage (databases), respectively. For instance, the medical terminology SNOMED CT with about 300 000 concepts is represented in OWL 2 EL (in $\mathcal{EL}^{++}$) and the QuOnto reasoner scales up to millions of tuples also with operational databases (Calvanese et al., 2010) and using a minor variant of OWL 2 QL, $DL\text{-}Lite_{\mathcal{A}}$, that has more features and adheres to the unique name assumption[9] (Calvanese et al., 2009). OWL 2 RL is inspired by Description Logic Programs and pD* and has special features to handle forward-chainging rules, yet restricts the OWL 2 DL syntax to keep decidability. Trade-offs between expressiveness of the ontology languages are illustrated and discussed in (Keet and Rodríguez, 2007). For instance, with OWL 2 DL one can avail of the following features in the ontology that OWL 2 QL does not have: role chaining, qualified number restrictions, enumerated classes, covering constraint over concepts, and transitivity on simple roles. On the other hand, with the leaner OWL 2 QL one can obtain similar performance as with relational databases thanks to the low computational complexity of the language (Calvanese et al., 2009), whereas for OWL 2 DL one never can achieve that (Motik et al., 2009a). Overall, each language has its advantages and disadvantages, and it would be useful to have an automated semantic transformation between any one of them. However, at the time of writing, no freely available applications exist that lets a modeller seamlessly and transparently change one ontology language for another for a given OWL 2-formalised ontology.

Given that the OWL 2 development came in response to experiences from OWL implementers, it is not unthinkable that an "OWL 3" will emerge, depending experiences gained with OWL 2. There is a 'language game' between modellers and logicians and a new list of problems has been made public (Schulz et al., 2009). Some of the issues Schulz and co-authors raised, however, can already be addressed in a DL language and a suitable automated reasoner, but they require various extensions to the OWL languages.

3.4.3 Extensions to the standard languages

The various challenges for representing subject domain semantics more precisely have resulted in development of logic-based languages in many directions. Limiting ourselves to DL languages, for they may have the greatest chance of being incorporated in some way into an "OWL 3", or at least being compatible with the standard OWL languages, then one can identify the work in progress into three distinct dimensions:

- Uncertainty: statements are true or false, but *due to lack of knowledge* we can only estimate to which probability / possibility / necessity degree they are true or false; e.g.: it rains in Galway or it does not, and the probability / possibility / necessity degree that it rains in Galway is 0.81;
- Vagueness: statements involve concepts for which *there is no exact definition*, such as tall, small, close, far, cheap, expensive; hence the statements are true to some degree, whereby the values are taken from a truth space, be it a multi-valued truth-space as in fuzzy logic, or the approximations of rough sets; e.g., "A Swedish citizen is very tall to degree 0.78";
- Temporal: instead of having wieldy ternary predicates including time t, one can include more succinct notions of time with the *until* and *since* operators, as well as the Allen temporal relations (e.g., $before$, $while$) using some version of Linear Temporal Logic or Computational Tree Logic. One then can represent information such as "Brain concussion after loss of consciousness" and "Caterpillar transformed into Butterfly" with the intended semantics and reason over such temporal knowledge.

There are languages and tools for probabilistic and fuzzy extensions to both several DL and OWL languages (Lukasiewicz and Straccia, 2008; Straccia., 2008). The probabilistic OWL language itself—and in particular $(\psi \mid \phi)[l, u]$ which means "generally, if an object belongs to ϕ, then it belongs to ψ with a probability in $[l, u]$"—can already solve Schulz et al.'s (2009) problem of representing "Hepatitis hasSymptom Fever in most *but not all* cases": by default, hepatitis has as symptom fever, so that we have in the ontology $(\exists \texttt{hasSymptom.Fever} \mid \texttt{Hepatitis})[1, 1]$.

Rough and fuzzy-rough DL and OWL-based ontology languages are introduced by, most notably, Bobillo and Straccia (2009) and Jiang et al. (2009). However, the little experimentation that has been carried out revealed that work is yet to be done to make it usable (Keet, 2010).

There are promising initial results in the direction of temporal logics: there are more and less expressive temporal DL languages, such as $\mathcal{DLR}_{\mathcal{US}}$ and *TDL-Lite* (Artale et al., 2002, 2007b; Lutz et al., 2008), its demonstrated usefulness for part-whole relations (Artale et al., 2008), and there is an OWL with temporal classes using a metric temporal logic (Keberle, 2009) for which there is an extension to the Pellet reasoner, OWL-MeT[10], to reason over the ontology. Such logics can, to a greater or lesser extent, deal with typical examples of SNOMED CT's "Bypass after grafting" (Schulz et al., 2009) and that Macrophage is a transformation of Monocyte (Keet, 2009). However, it is not clear yet if, and if so which, of these approaches will reach the maturity to be fully deployed in ontology development environment (ODE) tools.

Last, mereotopology and *spatial* logics, including the RCC set of spatial relations (Randell et al., 1992), may be of use. It is already known that RCC8 cannot be represented fully in OWL and trade-offs are under investigation (e.g., by Grütter and Bauer-Messmer (2007) regarding environmental data of the Swiss Alps). Any other extensions with respect to DLs-as-ontology-language, such as notions of belief revision, are in even earlier stages of research. Notwithstanding this, one always has the option to choose a non-DL logic as ontology language to meet one's needs (given that there are first order logic model checkers and theorem provers, such as Prover9 and Isabelle), but this comes at the cost of undecidability and performance losses.

3.5 Reasoning services

Automated reasoning services for ontologies can be grouped into three main categories: the DL-based reasoning services that gained prominence especially in the context of the Semantic Web, ontological reasoning services that aid the modeller to represent the subject domain semantic better, and reasoners that use a variety of different technologies to solve reasoning scenarios that the common DL-based reasoners do not address. In this section, we describe the high-level aspects as to *what* they do and what their purposes are.

3.5.1 Description logics-based reasoning services

The DL-based reasoning services fall into three categories, which are available to a greater or lesser extent due to features or limitations of the language and the ODEs that present the inferences.

1. 'Standard' reasoning services for ontology usage: satisfiability and consistency checking, taxonomic classification, instance classification;
2. 'Non-standard' reasoning services to facilitate ontology development: explanation/justification, glass-box reasoning, pin-pointing errors, least-common subsumer;
3. Querying functionalities, such as epistemic and (unions of) conjunctive queries;

Examples that clearly demonstrate the benefit of instance classification to the point of discovering (more precisely: deriving) novel biological knowledge and saving the researcher laboratory equipment and time are Wolstencroft et al. (2007) and Bandini and Mosca (2006), respectively. The other standard reasoning services are useful for ontology development, in particular when one chooses an expressive language or has a large ontology. However, if the reasoner finds unsatisfiable or inconsistent concepts, as a consequence, it may be that there are many more concepts unsatisfiable so that it becomes difficult to find the root of the problem. To this end, the 'non-standard' reasoning services have been developed, and the notions of pinpointing errors and explaining the entailments in particular (Brandt and Turhan, 2001; Horridge et al., 2008; Parsia et al., 2005).

Query functionalities logically and technically are part of the reasoning services, where the focus is not mere string matching ("find me [concept/role] x") but, e.g., "for the Firmicutes, retrieve the organisms and their genes that have a GC3 contents higher than 80" that uses a comparison operator (Calvanese et al., 2010), "is concept x part of concept y?" where x and y do not necessarily stand in a direct parthood relation, querying under incomplete information (Calvanese et al., 2009), or, e.g. "all devices that assist with upper limb mobility" that requires first an evaluation over the type-level ontology to find the type of devices that have declared an object property, say, *assistsWith* that has as range *UpperLimbMobility* and then to retrieve only those instances from the data (Alberts et al., 2008; Keet et al., 2008).

More requirements for reasoning services for use with bio-ontologies have been identified (Keet et al., 2007), which are partially satisfied by the extant reasoning services, partially in the pipeline (such as hypothesis testing with usable linking of data to an ontology), and partially belong to what can be dubbed *onto*logical reasoning services.

3.5.2 Ontological reasoning services

With ontological reasoning services we denote those algorithms that do contribute to ontological correctness of the knowledge represented in the ontology, but which are not part of the reasoning services for the logic-based ontology language in which the ontology is represented, hence, the obtained knowledge cannot be derived from that logical theory in, say, OWL. Thus far, we are aware of two such reasoning services, being OntoClean (Guarino and Welty, 2004) and the RBox reasoning service (Keet and Artale, 2008).

While the DL-based reasoners, such as FaCT++ and Racer, compute the subsumption hierarchy based on the declared object- and data properties, OntoClean computes the hierarchy based on the concepts' *metaproperties*. That is, for each concept in the hierarchy, one determines the kind of rigidity (rigid, semi-rigid, anti-rigid), identity, unity, and dependence, and then it sorts the concept in its appropriate order according to ontologically-motivated rules such as 'Given two concepts, p and q, when q subsumes p: if q is anti-rigid, then p must be anti-rigid' and that a rigid concept can subsume an anti-rigid one, but not vice versa. For instance, *Person* (rigid) can subsume *CollegeStudent* (anti-rigid), but not the other way around: it is always the case that each college student is also an instance of person, but it is not the case that every person is a college student.

The RBox reasoning service ensures that the role hierarchy is modelled correctly with respect to the subject domain. Whereas from a pure logic-and-reasoning standpoint, a hierarchy of relations (OWL object properties) is declared as given assertions and assumed to be correct, the RBox reasoning service checks if the sub-relations have their domain and range restriction such that these concepts are equal or subsumed by the domain and range of its super-relation. This is useful in particular for hierarchies of part-whole relations, but one can use it with other object relation hierarchies as well; e.g., `r_location` has as domain and range `region` and `unit` has domain `region` and range `measurement-unit`, then when the relation hierarchy has the assertion `unit` $\sqsubseteq$ `r_location`, the RBox reasoning service verifies that `measurement-unit` $\sqsubseteq$ `region` holds in the concept hierarchy (as is the case in the OWL-ized DOLCE-lite Plus) and not vice versa.

3.5.3 Other technologies

Although the Semantic Web technologies are gaining momentum for ontology engineering, its technologies are modular in the sense that one can, e.g., swap a DL-based reasoner for another one that takes an OWL file as input. We briefly mention three different directions for alternative reasoning technologies.

While it is possible to carry out instance classification with an OWL ontology (Wolstencroft et al., 2007), one has to declare the domain and range of each OWL object property, have defined concepts for more classification, and not too much data. If those constraints cannot be met, then data clustering techniques for instance classification can be a viable option to explore, be they ontology-enhanced (Zhou et al., 2005) or not.

There are reasoners for probabilistic ontologies, such as Da Costa and Laskey's (2006) approach that combines first order logic with Bayesian networks to reason over the probabilistic OWL PR-OWL.

Constraint programming is already being deployed for reasoning over conceptual data models (Cabot et al., 2008), and the step to do so for ontologies as well is relatively minor then. Possible advantages to have a constraint-based reasoner are that one already can have ternary relations and reason with temporal relations.

This concludes the brief exposé of ontology development parameters that will be assessed on their dependencies in the next section.

		OWL Language					Ontology reuse		
	SKOS	**2 QL**	**2 EL**	**2 DL**	**DL**	**Extensions**	**foundational**	**reference**	**domain**
Purpose ⇓									
1. Query data	−	+	−	−	−	+	−	−	±
2. Database integration	+	+	+	−	−	±	±	±	+
3. Integration / record navigation	+	+	+	−	−	−	−	±	+
4. Part of scientific discourse	−	−	−	+	+	+	+	+	+
5. Web services orchestration	−	−	+	±	+	−	±	+	+
6. ODIS	±	+	+	−	±	±	±	+	+
7. ontoNLP	+	+	+	−	±	−	±	+	+
8. Science	−	−	−	+	±	+	+	+	−
9. Tutorial ontology	−	−	−	+	+	±	−	−	+
Reasoning services ⇓									
1. Standard	−	±	±	+	+	+			
2. Non-standard	−	±	±	+	+	−			
3. Querying	−	+	+	−	−	±			
4. Ontological	+	+	+	+	+	+			
Bottom-up ⇓									
1. Other KR/CM	−	±	±	+	+	−			
2. DB reverse	−	±	±	+	+	−			
3. Textbook models	−	−	±	+	+	+			
4. Thesauri	+	±	+	−	−	−			
5. Other semi-structured	±	±	+	−	−	−			
6. Text mining	+	±	+	−	−	−			
7. Terminologies	+	±	+	−	−	−			
8. Tagging	+	±	+	−	−	−			
Ontology reuse ⇓									
1. Foundational	−	−	−	+	±	−			
2. Reference	−	±	±	+	+	−			
3. Domain	±	±	+	+	+	−			

Table 1 Basic cross-matching between realistic combinations of parameters. The more complex dependencies, such as the interaction between purpose, language, and reasoning service, can be obtained from traversing the table (*purpose* ↔ *language* and *language* ↔ *reasoning services*), likewise for, e.g., the dependencies between purpose and bottom-up development, assessing *purpose* ↔ *language* and then *bottom-up* ↔ *language*.

4 Dependencies between parameters

Given the aforementioned parameters, one obviously has to choose something when developing an ontology. Here we are not interested in how the developer has to choose, but instead in the question *what are consequences of such choices?* For instance, given purpose 4 (ontology as part of scientific discourse), then which ontology language and which reasoning services would suit best? Or, given that one's bottom-up ontology development tool locks one into OWL 2 QL, then what can one use the ontology for? Answers to such questions are included in Table 1, where "–" is to be understood "discouraged or not possible", "±" as "might work" but, e.g., no tool is available yet to do it or another combination is more suitable to achieve the optimal results, and "+" as a "workable or good combination". Although each entry in the table deserves an explanation, we will illustrate the guiding principles and rationale, from which the others follow. This section is divided into dependencies due to computational challenges, ontologies and their subject domains, useful combinations, and an experimental assessment of extant ontologies and their developers.

4.1 Dependencies due to computational challenges

An important factor is the trade-off between expressiveness of the language and the things one can do with an ontology represented in the language; for a practical explanation, the reader is referred to Keet and Rodríguez (2007), and for a more rigorous treatise on computation and computability, one can consult, e.g., Hopcroft et al. (2001). For instance, it is computationally utterly intractable to pose (unions of) conjunctive queries over medium to large-sized OWL-DL and OWL 2 DL ontologies, especially when they are linked to data, whereas OWL 2 QL was designed for this purpose. This advantage of OWL 2 QL comes at the cost of expressiveness of the language (see §3.4.2), so representing much of a scientific theory is not really possible; hence we obtain "+" in purposes 1-3 for 'simple' ontology languages, "+" for purposes 4, 5, 8 and 9 for expressive ontology languages, and a "+" for science (purposes 4 and 8) and very expressive languages (OWL 2 DL and extensions).

A downside of expressive ontology languages is that it is much easier to make mistakes when representing the subject domain and to lose oversight over the deductions. Therefore, besides the standard reasoning services, the non-standard reasoning services are important and, in fact, developed for exactly that purpose, so that there is a corresponding "+" in the table. The reasons why OWL 2 QL & non-standard reasoning services has "±" is that explanation in *DL-Lite* (Borgida et al., 2008) does not give the user as much and as useful information as with the others, because *DL-Lite* is a language of low espressiveness, and it has not been implemented yet to verify the theory. This holds even more so for OWL extensions, given that no non-standard reasoning services have been developed even in theory (therefore "–"). In this light, it might seem curious to have all "+" for the ontological reasoning services intersected with the language. However, they are orthogonal to the DL-based reasoning services and the item lumps together OntoClean and the RBox reasoning service that are useful in different scenarios. OntoClean gains importance especially with taxonomies when few, if any, object- and data properties have been described (thereby rendering meaningless the standard subsumption reasoning with DL-based reasoners) so that one can avail of at least the ontological subsumption reasoning to have a comparatively good taxonomy. The RBox reasoning service assumes object properties have been declared in the RBox and have been used widely in the ontology, which is typical for ontologies that are represented in an expressive language.

Looking at language extensions together with ontology reuse, there are "–" because there are hardly any real ontologies beyond toy examples. This does not mean that theoretically it is impossible, but one has to note that the more languages one uses for representing ontologies, the more challenging ontology reuse will become and the more it will be a case of ontology re-engineering (approximation, extensions, simplification etc.). For the same reason, the combination of other representation formalism and conceptual models together with OWL extensions have received a "–"; some theoretical results have been obtained with a temporal EER (Artale et al., 2007c) and fuzzy UML (Ma and Yan, 2007), but they are not implemented even in proof-of-concept tools in order to exploit such conceptual models for bottom-up development. In fact, the former has a DL foundation and can be seen in the light of DL or OWL extensions being transferred to conceptual modeling instead of vice versa. Nevertheless, such language extensions are important for the formalisation of the vocabulary of textbook models as bottom-up strategy ("+" in the table). For instance, the so-called Central Dogma in genetics is normally pictured in one figure where arrows denote the temporal notion of $after$ in the linear flow of time from transcription of DNA to translation into the protein, and likewise regarding the arrows in chemical reactions that can map onto, among others, "next time" ($\oplus$) for an instantaneous reaction, "at some time in the future" ($\Diamond^+$) for equilibria, or "at all times in the future" ($\Box^+$) for irreversible reactions, as well as the earlier mentioned (§3.3) STELLA models.

Other theoretical and technical challenges that make it practically a daunting task to develop an ontology is the 'overkill' of using an expressive ontology language, such as OWL 2 DL, with notoriously underspecified ontology-like 'simple' but large artifacts such as thesauri, terminologies, and tagging for use in bottom-up ontology development; hence, receiving a "–". For instance, AGROVOC[11] has about 30000 "descriptors" and the NCI metathesaurus contains 400,000 concepts mapped to 3,600,000 terms with 17,000,000 relationships, for which the OWL 2 DL algorithms were not built. Instead, it is appropriate to use scalable languages such as SKOS and OWL 2 EL, and, if the thesaurus is not only transformed into an ontology but also must be linked to large amounts of data, then OWL 2 QL might be of use (which, however, will blow up the encoding of the ontology due to the restrictions put on the usage of the concept constructors and relations). Initial results have been obtained to also provide a tractable mechanism to link OWL 2 EL to large amounts of data (Lutz et al., 2009; Baader et al., 2010), which might gain prominence as usage scenario in the near future. Notwithstanding the latter, it has been observed that thesauri tend to have linguistic or lexical aspects interwoven in the source (Soergel et al., 2004; Wielinga et al., 2001), which can be an advantage in particular for the purpose of an ontoNLP application or record navigation.

Thus, there are both theoretical limitations following from computability and engineering ones that concern gaps in the availability or functionality of ontology engineering tools, which make certain combinations practically feasible or prohibitively expensive in the light of the required resources.

4.2 Ontologies and subject domains

Different from theoretical and technical limitations, are the "–" for tutorial ontology (purpose 9) and reuse of foundational and reference ontologies. The principal reason is that foundational ontologies tend to be rather complex and not of immediate end-user (domain expert modeller) interest, thereby posing a hurdle in getting oneself acquainted with (logic-based) ontologies and domain experts and software engineers generally have more affinity with topics such as pizzas, wines, and university structures (at least initially).

A different assumption underlies the "–" crossing science as purpose and reuse of domain ontologies: there is not supposed to be any overlap between someone else's domain ontology $\mathcal{O}_{other}$ and one's own envisioned ontology ${}_{our}\mathcal{O}$, but a *complementarity* instead, be it through coordinated development and modularization of the subject domains upfront (Smith et al., 2007) or an extension of one of the branches of $\mathcal{O}_{other}$. This assumption follows directly from the reality-based approach (§2.1) that says that there can be only one ontology representing reality. Complementarity and orthogonality may be assumed for at least the life sciences, but it does not necessarily hold for social sciences and humanities, where competing paradigms and theories go in parallel. However, also for the latter domain it is prudent to keep the competing theories in separate ontologies, lest one ends up with an inconsistent ontology for this reason. Regarding complementarity, extension of $\mathcal{O}_{other}$ with the subject domain one wants to develop an ontology for is feasible and normally welcomed by developers of $\mathcal{O}_{other}$. This is different for extraction of a reusable section or a module from $\mathcal{O}_{other}$ to use in ${}_{our}\mathcal{O}$, because it is not trivial to demarcate the module and practically extract it, and it requires more careful versioning and maintenance management of both $\mathcal{O}_{other}$ and ${}_{our}\mathcal{O}$. Efforts have gone into automating the extraction of "sub-ontologies" (e.g., Konev et al., 2008), but they focus on finding sections that are isolated from the rest due to absence of connecting axioms, which may not necessarily coincide or overlap with the section that represents the domain of interest for the ontology developer. Hence, thus far, upfront modularization can increase the potential for reusability of, in particular, domain ontologies.

The values in the table for ontology reuse and the ontology language are motivated by practical aspects and characteristics of the ontologies. Foundational ontologies are certainly not represented in SKOS and not yet in OWL 2 QL or OWL 2 EL either, but instead in a very expressive language. This does not mean one cannot slim the foundational ontologies further to, say, a DOLCE ultra-ultra-lite (ultra-lite already exists), but it has not been done yet, thereby delaying the possibility to reuse a much simplified foundational ontology in a domain ontology intended for runtime usage. The same line of argumentation holds for reference ontologies in OWL 2 QL or OWL 2 EL, because these languages are intended for scalable implementations and not to represent a fairly generic subject domain as comprehensive as possible in an expressive language.

Last, the "–" in the extensions column is due to the limited knowledge about the extensions and limited availability of tools to use them in ontology development. Comparatively, the temporal extension is most requested for foundational ontologies and is currently incorporated in the (first order logic) characterisation with ternary predicates (DOLCE, GFO) or in the OWL ontology described in the comment field (BFO). Another prohibiting factor is that when a potentially reusable ontology is available, it may be with the 'wrong' extension; e.g., a probabilistic one whereas one wants to use it with vagueness predicates, or its fuzzy predicates adhere to Lukasiewics semantics whereas for one's own ontology Zadeh semantics suffices. In general, the issues that are raised by a mismatch between the language in which $\mathcal{O}_{other}$ is represented and the intended language for ${}_{our}\mathcal{O}$ are under-investigated and insufficient tools are available to (semi-)automatically go from one representation language to another.

4.3 Useful combinations

While many combinations are theoretically possible, it may have become clear from the previous two sections why some parameters do, or do not, go together. To synthesise those considerations, three typical combinations of purpose, language, reasoning services, bottom-up development, and ontology reuse are outlined in this section.

A scenario that is popular for ontologies in science, be it as part of an experiment or for representing a scientific theory, is a mostly top-down approach that reuses at least one foundational ontology, represents the subject domain in a (very) expressive ontology language (at least OWL DL), and uses both standard- and non-standard reasoning services as part of the ontology and its development process. Bottom-up activities, if any, are separated from the main ontology development process and used only to inform the modeller about possible terms that may denote universals (concepts) or relationships, which will be added in an semantically enriched way only if deemed appropriate after careful analysis. In addition, once the recent research outcomes regarding uncertainty, vagueness, and temporal logics become more widely known among the developers, one may expect that ontologies will be enriched with these dimensions; the latter will fill a known current gap, and the former two may be interesting in particular in the scientific method for hypothesis testing about the theory itself.

At the other end of the spectrum and from an engineering perspective, bottom-up ontology development using relational databases and their conceptual data models have appeal, despite that it is a non-trivial task. Although conceptual modelling languages are at least ExpTime (UML class diagrams) or even undecidable (ORM) (Berardi et al., 2005; Keet, 2007), the simplifications pushed through in the RDBMS implementations make it easier to remain with a much less expressive language, such as $DL\text{-}Lite_{\mathcal{A}}$ or even OWL 2 QL, which, in turn, fits well with Ontology-Based Data Access (OBDA)[12], powerful querying of both the ontology and the associated data, and light-weight data integration. Hence, an ontology-driven information system at run-time. In such a scenario, however, there will be little opportunity for reuse of ontologies for two main reasons: (a) at least the foundational and reference ontologies, if not also any (section of a) domain ontology, will have to be slimmed first (see §4.2), and (b) incorporating such ontologies in whole or in part into $_{our}\mathcal{O}$ increases the mismatch between the knowledge in the ontology and the data in the database or blows up the mappings between the terms in the ontology and queries over the database, which not only may affect the user experience when querying the ontology but also puts a larger burden on the system developer to design the implementation (Keet et al., 2008).

The third scenario is a mixture of bottom-up ontology development using thesauri, vocabularies (including OBO ontologies), NLP tools for ontology learning and population, Digital Libraries, and record or entity navigation-based approaches, for which SKOS and OWL 2 EL, and to a lesser extent possibly also OWL 2 QL, are suitable. Due to the low expressiveness of the language, one cannot obtain much from the standard reasoning services, and bothering oneself with top-down issues if an entity is an amount of matter or a physical object is not very relevant for linked data (that is, at present it is not perceived to be essential to include). In addition, it is nigh on impossible to obtain more subject domain semantics from the chosen sources without a full manual analysis.

Having outlined the more comprehensive combinations, this does not imply other combinations are strongly discouraged or even impossible (a fourth scenario will be discussed in Section 5); it just means that other combinations will require relatively more effort in ontology development where more theoretical and engineering hurdles have to be overcome.

4.4 Experimental assessment

Besides theoretical motivations for dependencies between ontology design parameters, we have examined a random selection of 17 bio-ontologies on their purposes, their subject domain, languages and so forth and have conducted a survey among ontology developers (11 respondents with another 10 ontologies). The hypotheses, methodology, materials, results, and discussion of the survey are described in a separate technical report (Alberts et al., 2008); due to space limitations, we will summarise only those outcomes that are directly relevant to the design parameters. A simplified list of dimensions of and about an ontology that was used to categorise the selection of bio-ontologies and asked about in the survey is included in the appendix. The categorisation of the ontologies according to these dimensions and their language as identified with the DL Metrics feature in Protégé is shown in Table 2. Note that recently the TONES Ontology Repository[13] has a longer list of ontologies (232 ontologies d.d. 19-2-2010), which follow the same pattern of type of ontology and the language used as the sample in the table.

Synthesising the assessment, most of the ontologies examined do not use the full expressivity of OWL-DL or OWL 2 DL. Among these ontologies, expressiveness of the ontology language that was used to represent the ontology correspond roughly to OWL 2 EL or QL, which suffice for the majority of bio-ontologies that have as aim data integration and linking to data sources. A definite trend can be observed that those ontologies generally use a much less expressive language compared to those focussed solely, or primarily, on representing type-level knowledge of a scientific theory. It is not clear why this is the case: do those ontology developers use a simpler language because of the focus on run-time usage of the ontologies linked to data, or do they not need a more expressive language in the first place? Curiously, there were discrepancies between the language features that were actually used in the ontology and the perceived requirements for language features selected by the survey respondents. This goes in both directions, i.e., where more is requested than is used and that more features are used than were requested. Given that selecting an ontology language is important for all four other design parameters, it deserves further investigation how to overcome this mismatch.

Further, inspection of the domain ontologies revealed that few of them reuse other ontologies, with the exception of BioTop, Microbial Loop, ADOLENA, and some of the OBO Foundry ontologies, but even they use only a highly simplified version of DOLCE or BFO and the Relation Ontology. The FMA, OBI, and BioPax are reference ontologies themselves that possibly could have been reused in whole or in part in the Human Developmental Anatomy Ontology, MGED and NMR, and PPI, respectively, but were not. Given that the survey did not include a question about ontology reuse, one can only speculate about the practical motivations in addition to the theoretical ones described in §4.2, and we are conducting experiments with modellers to understand better what the practical motivations for (non-)reuse are. Nevertheless, the history of the ontology may provide some explanation: e.g., the SELEX, HGT, and e-lis ontologies have a background in bottom-up ontology development having used relational databases and conceptual modelling as sources, and the Disease Ontology was based upon (and mapped to) medical billing codes, and therefore its developers may not have wanted to bother themselves with foundational and reference ontologies.

5 Discussion

Reflecting on the synthesis of the review on development of ontologies, the design parameters, and their dependencies, one is still left with an impression that they are largely 'soft' issues for which it is not

Ontology	Characterizing DL	Ontology type
Foundational Model of Anatomy, Lite[14]	N/A*	a, c, h, i, l, n
ProPreO	$\mathcal{SHOIN}(D)$	b, d, g, i, k
Ontology for biomedical investigations[15]	$\mathcal{SHOIN}(D)$	b, c, g, i, l, n
BioPax	$\mathcal{ALCHON}(D)$	a, c, e, i/j, l, n
Cell Cycle Ontology	$\mathcal{SIN}(D)$	a, d, g, j, k, n
HistOn	$\mathcal{ALCHIF}(D)$	ab, d, g, j, k, o
NMR Ontology[16]	$\mathcal{SH}(D)$	ab, c, d, g, i, k
BioTop[17]	$\mathcal{SH}$	a, d, e, j, l, o
MGED Ontology	$\mathcal{ALEOF}(D)$	b, c, g, j, k, m
Human Developmental Anatomy Ontology[18]	$\mathcal{ALEOF}(D)$	ab, c, f, i, k, m
Microbial Loop[19]	$\mathcal{ALCHI}$	a, d, g, j, k, o
Cell Type Ontology	$\mathcal{ALE}(D)$	b, c, h, j, k, m
Gene Ontology[20]	$\mathcal{ALE}(D)$	b, c, h, i, k, m
Protein-Protein Interaction Ontology[21]	$\mathcal{ALE}(D)$	b, c, h, i, k, m
Mammalian Phenotype Ontology[22]	$\mathcal{AL}(D)$	b, c, f, i, k, m
Disease Ontology[23]	$\mathcal{AL}$	b, c, h, j, k, m
FungalWeb[24]	$\mathcal{FL}_0$	b, cd, g, j, k, m
"OBO Foundry; particular GO, PATO, SO, Anatomy, .."		ab, c, h, i, k, m
GO	$\mathcal{ALE}(D)$	
PATO*	$\mathcal{AL}$	
SO**	$\mathcal{SHI}$	
Cell Cycle Ontology	$\mathcal{SIN}(D)$	a, b, d, h
Microbial Loop	$\mathcal{ALCHI}$	a, d, g, j, k, o
iCAPTURer	in development	a, d, f, g, k
"OAT"	N/A	a, b, g, j
e-lis	$\mathcal{ALCHN}$	b, d, g, i, k, o
"cars (automotive domain)"	$\mathcal{ALCOIF}(D)$	d, g, j, k, o
ADOLENA	$\mathcal{SHIQ}$	a, b, d, g, j, k, o
HGT	legacy representation (ORM2, in NORMA)	b, d, h, j, k, o
SELEX	$\textit{DL-Lite}_{\mathcal{A}}$	b, d, g, j, k

*: too large to load in Protégé

Table 2 DL characterization of the expressivity of several bio-ontologies sorted in (approximate) decreasing order with respect to the complexity of the language in the first part of the table and in order of the survey respondents in the second part of the table, each with their type and purposes of those ontologies in the third column (see Appendix for the meaning of the letters in the right-most column).

unthinkable to find a counter-example and some of the borderline cases ("±") in Table 1 might change for the better in the future once more tools have been developed and more ontologies have been developed. Viewed from this angle, ontology development still seems more experimentation and skills than a scientifically underpinned endeavour. Nevertheless, most inherent challenges, such as the limitations due to computational complexity of the ontology languages and the challenges for reusing foundational ontologies in domain- and 'application ontologies', remain and have their scientific bases and argumentations as to why this is the case. Concerning the useful combinations (§4.3), it is difficult to establish why exactly they emerged: is it because there were tools to do it, or were the tools developed to fit those scenarios? For instance, an appealing fourth scenario is to develop a comprehensive ontology in a very expressive language for the sake of representing the subject domain as accurate and precise as possible, and subsequently to use various 'simplifiers' to have less expressive versions of the ontology in, say, OBO, OWL 2 EL, and OWL 2 QL, that can be used in various ontology-driven information systems. But (semi-)automated semantic simplifiers still do not exist, even though it has been requested for (e.g., by Keet et al., 2008), and carrying it out and maintaining it manually is rather prohibitive. This means a tool-pull to meet implementation scenarios. This also holds for, e.g., the creative "model violation" to narrow down the potentially interesting molecules from a large set to a few, i.e., instance classification where the aim is to have many inconsistent instances (Bandini and Mosca, 2006), and the use of ontologies in the Taverna workflow system (Goble et al., 2007). Conversely, although rules and ontologies, $\mathcal{EL}^{++}$, and *DL-Lite* do have a (very) small user-base, making them part of the OWL 2 standard as profiles amounts to a push for the large majority of ontology developers.

We return now to the questions from the introduction. The first question—which strategies and additional methods do developers employ when they do not adhere to a particular methodology—can be answered with: the possible combinations of the permutations in the five groups of design parameters. What the possible permutations at each step are, the choices the developers face, and if they can be categorised in types of choices, can be answered, first and foremost, with the list of structured parameters described in Section 3, and, indirectly, with the dependencies between them. Thus, on the question 'does a choice for one method, tool, language and so forth lock one into another', the answer is in the affirmative and summarised in Table 1. Then, regarding the last question—why do they go together or are exclusive, and can or should that be changed—the answer follows from the theoretical and practical assessment in Section 4. Many dependencies can be traced back to the ontology language with respect to the computational complexity and implicit ontological commitment, then to the subject domain, and to a lesser extent also the bottom-up source material. Put differently, purpose and reasoning services play second fiddle to the other three groups of parameters. While it is true that a purpose can lock one into a ontology language, it is the language that imposes the constraint, likewise that the reasoning services available depend on the language. This is not to say that at the commencement of ontology development one should first choose an ontology language, but merely that for whichever combination of permutations one chooses from the four other sets of parameters, success of development certainly will be enhanced or obstructed by the ontology language.

Last, and separate from the analysis presented in this paper, it may be useful to develop a software-supported procedure to select a workable combination of parameters. For instance, a developer selects the desired purpose and reasoning services and the software answers with a suitable language, or she decides on a purpose of the ontology and language so as to obtain a list with the available reasoning services. Perhaps additional functionality could be to provide also suggestions for the most appropriate tools, which indirectly will reveal gaps in software support for ontology development (e.g., explanation in $DL\text{-}Lite_{\mathcal{A}}$, simplifiers, temporal DL reasoners) that, when filled up, can increase the amount of "+" in Table 1 and thereby serve the ontologist better. This, in turn, may be extended with soft constraints to facilitate ontology development with modeller-oriented features of the tools, such as intelligent ontology visualisation and graphical querying to reduce the cognitive overload, and to find a minimal set of tools one has to become acquainted with to carry out the ontology development project.

6 Conclusions

To improve on the efficiency and effectiveness of development of, primarily, domain ontologies, we described five influential factors and their dependencies. These five main parameters are (i) nine types of purpose(s) of the ontology, (ii) what and how to reuse the three main types of existing ontologies, (iii) eight different types of approaches for bottom-up ontology development from other legacy sources, (iv) representation languages including the OWL and OWL species and extensions, and (v) four types of reasoning services. The dependencies between these parameters have been structured, and follow primarily from computational challenges—both theoretical and application support—and types of ontologies with their subject domain. The theoretical analysis was assessed against a random selection of ontologies and a survey among ontology developers, which revealed that they roughly adhere to the outcome of the theoretical analysis.

Future works is directed toward integrating the dependencies into ontology development methodologies and setting up a software-mediated guidance system that can make suggestions how to proceed with ontology development given particular requirements; thus, to make the 'soft' knowledge about ontology development more accessible and shorten the development process.

Appendix

The simplified list of dimensions of and about an ontology, which was used to categorise the selection of bio-ontologies in Table 2 and asked about in the survey, is as follows:

1. Attitude toward the ontology:
   - (a) representing reality at the type-level, say, as a way to represent a scientific theory;
   - (b) to link it to their data, hence, within an (immediate) application focus;
2. Aim of the ontology:
   - (c) ontologies as more or less official standardization effort;
   - (d) experimental ontologies that are part of doing science (be it science in a subject domain or within computer science and engineering);
   - (e) tutorial ontologies;
   - (f) ontologies for knowledge acquisition (e.g., machine learning, text mining, wisdom of the crowd);
3. History of the development of the ontology:
   - (g) development *de novo* and (aimed at) being formal ontologies;
   - (h) legacy ontologies that have been converted automatically into OWL or are/were thesauri that intend to be upgraded to passing for an ontology;
4. Characteristic of the developers:
   - (i) within, e.g., the OBO Foundry scope designated as a candidate ontology, or some other consortium;

(j) developed independently from centralised efforts, i.e., in a particular research group, or by an individual;

5. Scope of the content of the ontology
   (k) domain ontology with Universe of Discourse a sub-discipline in biology, ecology, chemistry, biomedicine, healthcare;
   (l) reference or top-level ontology, such as DOLCE, GFO, BFO, FMA, BioTop;
6. Size and/versus expressivity of the language used for the ontology:
   (m) large but simple, i.e., mostly just a taxonomy without, or very few, properties (relations) linked to the concepts, where 'large' is, roughly, > 10000 concepts;
   (n) large and complex, where 'complex' includes rich usage of properties, defined concepts, and, roughly, requiring at least OWL-DL;
   (o) small and complex, where 'small' is, roughly, $<$ 250 concepts.

References

R. Alberts, D. Calvanese, G. D. Giacomo, A. Gerber, M. Horridge, A. Kaplunova, C. M. Keet, D. Lembo, M. Lenzerini, M. Milicic, R. Möller, M. Rodríguez-Muro, R. Rosati, U. Sattler, B. Suntisrivaraporn, G. Stefanoni, A.-Y. Turhan, S. Wandelt, and M. Wessel. Analysis of test results on usage scenarios. Deliverable TONES-D27 v1.0, TONES Project, Oct. 10 2008.

D. Alexopoulou, T. Wächter, L. Pickersgill, C. Eyre, and M. Schroeder. Terminologies for text-mining; an experiment in the lipoprotein metabolism domain. *BMC Bioinformatics*, 9(Suppl 4):S2, 2008.

A. Amoroso, G. Esposito, D. Lembo, P. Urbano, and R. Vertucci. Ontology-based data integration with MASTRO-I for configuration and data management at SELEX Sistemi Integrati. In S. Gaglio, I. Infantino, and D. Saccà, editors, *SEBD*, pages 81–92, 2008.

A. Artale, E. Franconi, F. Wolter, and M. Zakharyaschev. A temporal description logic for reasoning about conceptual schemas and queries. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the 8th Joint European Conference on Logics in Artificial Intelligence (JELIA-02)*, volume 2424 of *LNAI*, pages 98–110. Springer Verlag, 2002.

A. Artale, D. Calvanese, R. Kontchakov, V. Ryzhikov, and M. Zakharyaschev. Reasoning over extended ER models. In *ER-07*, volume 4801 of *LNCS*, pages 277–292. Springer, 2007a.

A. Artale, R. Kontchakov, C. L. nad Frank Wolter, and M. Zakharyaschev. Temporalising tractable description logic. In *Proc. of the 14th International Symposium on Temporal Representation and Reasoning (TIME-07)*, 2007b. Alicante, June 2007.

A. Artale, C. Parent, and S. Spaccapietra. Evolving objects in temporal information systems. *Annals of Mathematics and Artificial Intelligence*, 50(1-2):5–38, 2007c.

A. Artale, N. Guarino, and C. M. Keet. Formalising temporal constraints on part-whole relations. In G. Brewka and J. Lang, editors, *11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*, pages 673–683. AAAI Press, 2008. Sydney, Australia, September 16-19, 2008.

A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyaschev. Dl-lite without the unique name assumption. In *Proc. of the 22nd Int. Workshop on Description Logic (DL 2009)*, volume 477 of *CEUR-WS*, 2009. http://ceur-ws.org/.

I. N. Athanasiadis, A.-E. Rizzoli, S. Janssen, E. Andersen, and F. Villa. Ontology for seamless integration of agricultural data and models. In F. Sartori, M. ngel Sicilia, and N. Manouselis, editors, *Proc. of Third International Conference on Metadata and Semantic Research (MTSR'09)*, volume 46 of *CCIS*, pages 282–293. Springer, 2009. Oct 1-2 2009 Milan, Italy.

F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logics Handbook – Theory and Applications*. Cambridge University Press, 2003.

F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. *Journal of Applied Logic*, 5(3):392–420, 2007.

F. Baader, M. Bienvenu, C. Lutz, and F. Wolter. Query and predicate emptiness in description logics. In *Proc. of KR'10*, 2010.

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

F. Belleau, M. Nolin, N. Tourigny, P. Rigault, and J. Morissette. Bio2rdf: Towards a mashup to build bioinformatics knowledge system. *Journal of Biomedical Informatic*, 41(5):706–716, 2008.

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

C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia – a crystallization point for the web of data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7:154165, 2009.

C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems, Special Issue on Linked Data*, in press.

F. Bobillo and U. Straccia. Supporting fuzzy rough sets in fuzzy description logics. In C. Sossai and G. Chemello, editors, *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'09)*, volume 5590 of *LNCS*, pages 676–687. Springer, 2009.

A. Borgida, D. Calvanese, and M. Rodríguez-Muro. Explanation in the DL-Lite family of description logics. In *Proc. of the 7th Int. Conf. on Ontologies, DataBases, and Applications of Semantics (ODBASE 2008)*, volume 5332 of *LNCS*, pages 1440–1457. Springer, 2008.

S. Borgo and C. Masolo. Foundational choices in DOLCE. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 361–381. Springer, 2 edition, 2009.

S. Brandt and A.-Y. Turhan. Using non-standard inferences in description logics - what does it buy me? In *Proceedings of the KI Workshop on Applications of Description Logics 2001 (KIDLWS2001)*, 2001. Vienna, Austria.

S. Brockmans, P. Haase, L. Serafini, and H. Stuckenschmidt. Formal and conceptual comparison of ontology mapping languages. In H. Stuckenschmidt, C. Parent, and S. Spaccapietra, editors, *Modular Ontologies*, volume 5445 of *LNCS*, chapter 10, pages 267–291. Springer-Verlag, 2009.

J. Cabot, R. Clarisó, and D. Riera. Verification of UML/OCL class diagrams using constraint programming. In *Model Driven Engineering, Verification, and Validation: Integrating Verification and Validation in MDE (MoDeVVA 2008)*, 2008.

D. Calvanese, G. D. Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, and R. Rosati. Ontologies and databases: The DL-Lite approach. In S. Tessaris and E. Franconi, editors, *Semantic Technologies for Informations Systems - 5th Int. Reasoning Web Summer School (RW 2009)*, volume 5689 of *LNCS*, pages 255–356. Springer, 2009.

D. Calvanese, C. Keet, W. Nutt, M. Rodríguez-Muro, and G. Stefanoni. Web-based graphical querying of databases through an ontology: the WONDER system. In *Proceedings of ACM Symposium on Applied Computing (ACM SAC'10)*, page to appear. ACM, 2010. March 22-26 2010, Sierre, Switzerland.

K.-H. Cheung, H. R. Frost, M. S. Marshall, E. Prudhommeaux, M. Samwald, J. Zhao, and A. Paschke. A journey to semantic web query federation in the life sciences. *BMC Bioinformatics*, 10(Suppl 10): S10, 2009.

P. Cimiano, P. Haase, M. Herold, M. Mantel, and P. Buitelaar. Lexonto: A model for ontology lexicons for ontology-based nlp. In *Proc. of the OntoLex (From Text to Knowledge: The Lexicon/Ontology Interface) workshop at ISWC07*, 2007. Busan, South-Korea, Nov. 2007.

O. Corcho, M. Fernandez-Lopez, and A. Gomez-Perez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & Knowledge Engineering*, 46(1):41–64, 2003.

P. C. G. da Costa and K. B. Laskey. PR-OWL: A framework for probabilistic ontologies. In *Proceedings FOIS'06*, pages 237–249. IOS Press, 2006.

R. Denaux, A. Cohn, V. Dimitrova, and G. Hart. Towards modelling the intended purpose of ontologies: A case study in geography. In D. Kolas, N. Wiegand, and G. Berg-Cross, editors, *Proceedings of the Terra Cognita Workshop, collocated with the 8th International Semantic Web Conference (ISWC-2009)*, volume 518 of *CEUR-WS*, 2009. Washington DC, USA, October 26, 2009.

H. Dietze, D. Alexopoulou, M. R. Alvers, L. Barrio-Alvers, B. Andreopoulos, A. Doms, J. Hakenberg, J. Moennich, C. Plake, A. Reischuck, L. Royer, T. Waechter, M. Zschunke, and M. Schroeder. Gopubmed: Exploring pubmed with ontological background knowledge. In S. A. Krawetz, editor, *Bioinformatics for Systems Biology*. Humana Press, 2008.

M. Fernandez, A. Gomez-Perez, A. Pazos, and J. Pazos. Building a chemical ontology using METHONTOLOGY and the ontology design environment. *IEEE Expert: Special Issue on Uses of Ontologies*, January/February: 37–46, 1999.

A. Gangemi. Applying ontology design patterns to practical expertise: roles, tasks and techniques in the agricultural domain. In C. M. Keet and E. Franconi, editors, *CSBio Reader: extended abstracts of the 'CS & IT with/for Biology' seminar series*, volume 1, pages 47–57. KRDB Research Centre, Free University of Bozen-Bolzano, Dec. 2005.

Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

Gene Ontology Consortium. The Gene Ontology GO database and informatics resource. *Nucleic Acids Research*, 32(1):D258–D261, 2004.

C. Ghidini, B. Kump, S. Lindstaedt, N. Mabhub, V. Pammer, M. Rospocher, and L. Serafini. Moki: The enterprise modelling wiki. In *Proceedings of the 6th Annual European Semantic Web Conference (ESWC2009)*, 2009. Heraklion, Greece, 2009 (demo).

A. M. Gliozzo, C. Caracciolo, M. d'Aquin, M. Sabou, W. Peters, and J. Voelker. Results from experiments in ontology learning including evaluation and recommendation. Deliverable 7.3.1, NeoN Project, Dec. 15 2007.

C. Goble, K. Wolstencroft, A. Goderis, D. Hull, J. Zhao, P. Alper, P. Lord, C. Wroe, K. Belhajjame, D. Turi, R. Stevens, T. Oinn, and D. D. Roure. Knowledge discovery for biology with taverna. In C. Baker and H. Cheung, editors, *Semantic Web: Revolutionizing knowledge discovery in the life sciences*, pages 355–395. Springer: New York, 2007.

C. Golbreich and I. Horrocks. The OBO to OWL mapping, GO to OWL 1.1! In *Proc. of the Third OWL Experiences and Directions Workshop*, volume 258 of *CEUR-WS*, 2007. http://ceur-ws.org/.

B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.

T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

R. Grütter and B. Bauer-Messmer. Combining OWL with RCC for spatioterminological reasoning on environmental data. In *Third international Workshop OWL: Experiences and Directions (OWLED 2007)*, 2007. 6-7 June 2007, Innsbruck, Austria.

N. Guarino. Formal ontology and information systems. In *Proceedings of Formal Ontology in Information Systems (FOIS'98)*. Amsterdam: IOS Press, 1998.

N. Guarino and C. Welty. An overview of OntoClean. In S. Staab and R. Studer, editors, *Handbook on ontologies*, pages 151–159. Springer Verlag, 2004.

J.-L. Hainaut, M. Chandelon, C. Tonneau, and M. Joris. Contribution to a theory of database reverse engineering. In *Reverse Engineering, 1993., Proceedings of Working Conference on*, pages 161–170, May 1993.

T. Halpin and T. Morgan. *Information modeling and relational databases.* Morgan Kaufmann, 2nd edition, 2008.

L. Han, T. Finin, C. Parr, J. Sachs, and A. Joshi. Rdf123: a mechanism to transform spreadsheets to rdf. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI'06)*, 2006.

N. Henze, P. Dolog, and W. Nejdl. Reasoning and ontologies for personalized e-learning in the semantic web. *Educational Technology & Society*, 7(4):82–97, 2004.

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

J. R. Hobbs and F. Pan. An ontology of time for the semantic web. *ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing*, 3(1):66–85, 2004.

J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Pearson Education, 2 edition, 2001.

M. Horridge, H. Knublauch, A. Rector, R. Stevens, and C. Wroe. A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools. Technical Report 1.0, The University Of Manchester, 2004. URL `http://www.coode.org/resources/tutorials/ProtegeOWLTutorial.pdf`.

M. Horridge, B. Parsia, and U. Sattler. Laconic and precise justifications in OWL. In *Proc. of the 7th International Semantic Web Conference (ISWC 2008)*, volume 5318 of *LNCS*. Springer, 2008.

I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7, 2003.

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

P. Jaiswal, D. Ware, J. Ni, K. Chang, W. Zhao, S. Schmidt, X. Pan, K. Clark, L. Teytelman, S. Cartinhour, L. Stein, and S. McCouch. Gramene: development and integration of trait and gene ontologies for rice. *Comparative and Functional Genomics*, 3:132–136, 2002.

M. Jarrar, J. Demy, and R. Meersman. On using conceptual data modeling for ontology engineering. *Journal on Data Semantics: Special issue on Best papers from the ER/ODBASE/COOPIS 2002 Conferences*, 1(1): 185–207, 2003.

Y. Jiang, J. Wang, S. Tang, and B. Xiao. Reasoning with rough description logics: An approximate concepts approach. *Information Sciences*, 179:600–612, 2009.

N. Keberle. Temporal classes and OWL. In R. Hoekstra and P. F. Patel-Schneider, editors, *Proceedings of OWLED'09*, 2009.

C. M. Keet. Factors affecting ontology development in ecology. In B. Ludäscher and L. Raschid, editors, *Data Integration in the Life Sciences 2005 (DILS2005)*, volume 3615 of *LNBI*, pages 46–62. Springer Verlag, 2005. San Diego, USA, 20-22 July 2005.

C. M. Keet. Prospects for and issues with mapping the Object-Role Modeling language into $\mathcal{DLR}_{ifd}$. In *20th International Workshop on Description Logics (DL'07)*, volume 250 of *CEUR-WS*, pages 331–338, 2007. 8-10 June 2007, Bressanone, Italy.

C. M. Keet. Constraints for representing transforming entities in bio-ontologies. In R. Serra and R. Cucchiara, editors, *11th Congress of the Italian Association for Artificial Intelligence (AI*IA 2009)*, volume 5883 of *LNAI*, pages 11–20. Springer Verlag, 2009. Reggio Emilia, Italy, Dec. 9-12, 2009.

C. M. Keet. On the feasibility of description logic knowledge bases with rough concepts and vague instances. In *Proc. of DL'10*, CEUR-WS, 2010. 4-7 May 2010, Waterloo, Canada.

C. M. Keet and A. Artale. Representing and reasoning over a taxonomy of part-whole relations. *Applied Ontology*, 3(1-2):91–110, 2008.

C. M. Keet and M. Rodríguez. Toward using biomedical ontologies: trade-offs between ontology languages. In *AAAI 2007 Workshop Semantic eScience (SeS 2007)*, volume WS-07-11 of *AAAI Technical Report*, pages 65–68. AAAI, 2007. 23 July 2007, Vancouver, Canada.

C. M. Keet, M. Roos, and M. S. Marshall. A survey of requirements for automated reasoning services for bio-ontologies in OWL. In *Proceedings of the 3rd Workshop on OWL: Experiences and Directions (OWLED 2007)*, volume 258 of *CEUR-WS*, 2007. 6-7 June 2007, Innsbruck, Austria.

C. M. Keet, R. Alberts, A. Gerber, and G. Chimamiwa. Enhancing web portals with Ontology-Based Data Access: the case study of South Africa's Accessibility Portal for people with disabilities. In C. Dolbear, A. Ruttenberg, and U. Sattler, editors, *Proceedings of the Fifth OWL: Experiences and Directions (OWLED 2008)*, volume 432 of *CEUR-WS*, 2008. Karlsruhe, Germany, 26-27 October 2008.

J. Kim, Y. Gil, and V. Ratnakar. Semantic metadata generation for large scientific workflows. In *Proceedings of the 5th International Semantic Web Conference (ISWC'06)*, 2006. Athens, GA, USA.

B. Konev, C. Lutz, D. Walther, and F. Wolter. Semantic modularity and module extraction in description logics. In M. Ghallab, C. D. Spyropoulos, N. Fakotakis, and N. M. Avouris, editors, *Proceedings of 18th European Conference on Artificial Intelligence (ECAI'08)*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, pages 55–59. IOS Press, 2008. Patras, Greece, July 21-25, 2008.

L. Lubyte and S. Tessaris. Extracting ontologies from relational databases. In D. Calvanese, E. Franconi, V. Haarslev, D. Lembo, B. Motik, S. Tessaris, and A.-Y. Turhan, editors, *Proceedings of the 20th International Workshop on Description Logics (DL 2007)*, pages 387–395, 2007.

T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6(4):291–308, 2008.

C. Lutz, F. Wolter, and M. Zakharyaschev. Temporal description logics: A survey. In *Proc. of the Fifteenth International Symposium on Temporal Representation and Reasoning (TIME'08)*. IEEE Computer Society Press, 2008.

C. Lutz, D. Toman, and F. Wolter. Conjunctive query answering in the description logic el using a relational database system. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence IJCAI09*. AAAI Press, 2009.

Z. Ma and L. Yan. Fuzzy XML data modeling with the UML and relational data models. *Data & Knowledge Engineering*, 63(3):972 – 996, 2007. ISSN 0169-023X. 25th International Conference on Conceptual Modeling (ER 2006) - 'Four of the best papers presented.

J. S. Madin, S. Bowers, M. P. Schildhauer, and M. B. Jones. Advancing ecological research with ontologies. *Trends in Ecology & Evolution*, 23(3):159–168, 2008.

C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Ontology library. WonderWeb Deliverable D18 (ver. 1.0, 31-12-2003)., 2003. http://wonderweb.semanticweb.org.

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

B. Motik, B. C. Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 Web Ontology Language Profiles. W3c recommendation, W3C, 27 Oct. 2009a. http://www.w3.org/TR/owl2-profiles/.

B. Motik, P. F. Patel-Schneider, and B. Parsia. OWL 2 web ontology language structural specification and functional-style syntax. W3c recommendation, W3C, 27 Oct. 2009b. http://www.w3.org/TR/owl2-syntax/.

B. Parsia, E. Sirin, and A. Kalyanpur. Debugging OWL ontologies. In *Proceedings of the World Wide Web Conference (WWW 2005)*, 2005. May 10-14, 2005, Chiba, Japan.

A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *J. on Data Semantics*, X:133–173, 2008.

V. Presutti, A. Gangemi, S. David, G. A. de Cea, M. C. Surez-Figueroa, E. Montiel-Ponsoda, and M. Poveda. A library of ontology design patterns: reusable solutions for collaborative design of networked ontologies. NeOn deliverable D2.5.1, NeOn Project, Institute of Cognitive Sciences and Technologies (CNR), 2008.

D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning*, pages 165–176. Morgan Kaufmann, 1992.

M. Rodriguez-Muro, L. Lubyte, and D. Calvanese. Realizing Ontology Based Data Access: A plug-in for Protégé. In *Proc. of the Workshop on Information Integration Methods, Architectures, and Systems (IIMAS 2008)*. IEEE Computer Society, 2008.

G. Schreiber. Knowledge engineering. In V. Lifschitz, F. van Harmelen, and B. Porter, editors, *Handbook of Knowledge Representation*, pages 929–946. Elsevier, 2007.

S. Schulz, H. Stenzhorn, M. Boekers, and B. Smith. Strengths and limitations of formal ontologies in the biomedical domain. *Electronic Journal of Communication, Information and Innovation in Health (Special Issue on Ontologies, Semantic Web and Health)*, 3(1):31–45, 2009.

B. Smith. Ontology (science). In *Proc. of FOIS'08*. IOS Press, 2008.

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

B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. Goldberg, K. Eilbeck, A. Ireland, C. Mungall, T. OBI Consortium, N. Leontis, A. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, M. Shah, P. Whetzel, and S. Lewis. The OBO Foundry: Coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25 (11):1251–1255, 2007.

D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: the AGROVOC example. *Journal of Digital Information*, 4(4), 2004.

U. Straccia. Managing uncertainty and vagueness in description logics, logic programs and description logic programs. In *Reasoning Web, 4th International Summer School*, 2008.

M. C. Suarez-Figueroa, G. A. de Cea, C. Buil, K. Dellschaft, M. Fernandez-Lopez, A. Garcia, A. Gomez-Perez, G. Herrero, E. Montiel-Ponsoda, M. Sabou, B. Villazon-Terrazas, and Z. Yufei. Neon methodology for building contextualized ontology networks. NeOn Deliverable D5.4.1, NeOn Project, 2008.

K. Vila and A. Ferrández. Developing an ontology for improving question answering in the agricultural domain. In F. Sartori, M. Sicilia, and N. Manouselis, editors, *3rd International Conference on Metadata and Semantics (MTSR'09)*, volume 46 of *CCIS*, pages 245–256. Springer, 2009. Oct 1-2 2009 Milan, Italy.

B. J. Wielinga, A. T. Schreiber, J. Wielemaker, and J. A. C. Sandberg. From thesaurus to ontology. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*, pages 194–201, New York, NY, USA, 2001. ACM. ISBN 1-58113-380-4. doi: http://doi.acm.org/10.1145/500737.500767.

R. Witte, T. Kappler, and C. Baker. Ontology design for biomedical text mining. In C. Baker and H. Cheung, editors, *Semantic Web: revolutionizing knowledge discovery in the life sciences*, pages 281–313. Springer, 2007.

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

S. Zhang, O. Bodenreider, and C. Golbreich. Experience in reasoning with the foundational model of anatomy in OWL DL. In R. B. Altman, A. K. Dunker, L. Hunter, T. A. Murray, and T. E. Klein, editors, *Pacific Symposium on Biocomputing (PSB'06)*, pages 200–211. World Scientific, 2006.

Y. Zhou, J. A. Young, A. Santrosyan, K. Chen, S. F. Yan, and E. A. Winzeler. In silico gene function prediction using ontology-based pattern identification. *Bioinformatics*, 21(7):1237–1245, 2005.

Notes

[1] A scientific answer to the question why this is the case is outside the scope of this paper. Anecdotal evidence directs toward a complex answer, including, but not limited to, factors such as unfamiliarity that methodolgies exist (ignorance of the modeller), how to use a particular methodology (lack of training or documentation), perceived gaps in the methodology (inadequacy of the methodology, be it the theory or the software support for it), and that one anyhow knows how to model or insists on reinventing the wheel (arrogance of the modeller).

[2] http://swoogle.umbc.edu/

[3] http://owl.cs.manchester.ac.uk/repository/

[4] http://bioportal.bioontology.org/

[5] This was already observed earlier in Object-Role Modeling; see (Halpin and Morgan, 2008) for a recent overview and more comprehensive argumentation.

[6] http://www.imbi.uni-freiburg.de/biotop/

[7] http://www.ariadnegenomics.com/products/pathway-studio/

[8] http://ncim.nci.nih.gov/

[9] The unique name assumption (UNA) is the norm in information systems, whereas OWL does not adhere to the UNA. No UNA increases computational complexity (Artale et al., 2009) and results in different inferences compared to, e.g., relational database systems (see, e.g., the example for the Haskey featrue in the OWL standard).

[10] http://ermolayev.com/owl-met/

[11] A vocabulary of terms regarding agriculture, http://aims.fao.org/website/AGROVOC-Thesaurus/sub

[12] Realistically, however, the 'ontology' is much closer to a conceptual model than an ontology, hence, 'COnceptual MOdel-based Data Access' (COMODA) would be a more accurate term; see e.g. the experimental results with operational databases in (Alberts et al., 2008; Calvanese et al., 2010).

[13] http://owl.cs.manchester.ac.uk/repository/

[14] http://sig.biostr.washington.edu/projects/fm/AboutFM.html; the OWL version, FMA-lite, can be accessed through http://www.obofoundry.org/cgi-bin/detail.cgi?id=fma_lite.

[15] http://www.obofoundry.org/cgi-bin/detail.cgi?id=obi

[16] Descriptors relevant to the experimental conditions of the Nuclear Magnetic Resonance (NMR) component in a metabolomics investigation http://obo.sourceforge.net/cgi-bin/detail.cgi?nmr, with general home page at http://msi-ontology.sourceforge.net/

[17] http://www.imbi.uni-freiburg.de/biotop/

[18] http://obo.sourceforge.net/cgi-bin/detail.cgi?human-dev-anat-abstract

[19] http://www.meteck.org/supplDILS.html

[20] http://www.geneontology.org/; the go_daily-termdb.owl was examined last on 25 Sept. 2008.

[21] http://obo.sourceforge.net/cgi-bin/detail.cgi?psi-mi and its central development page at http://psidev.sourceforge.net/mi/xml/doc/user/index.html

[22] http://obo.sourceforge.net/cgi-bin/detail.cgi?mammalian_phenotype

[23] http://diseaseontology.sourceforge.net/

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