Towards a Standard for Heterogeneous Ontology Integration and Interoperability

Oliver Kutz[1]*, Till Mossakowski[1,2], Christian Galinski[3], and Christoph Lange[1,4]

[1] Research Center on Spatial Cognition, University of Bremen, Germany
[2] DFKI GmbH Bremen and University of Bremen, Germany
[3] Infoterm, Vienna
[4] Computer Science, Jacobs University, Bremen, Germany

Abstract. Even though ontologies are widely being used to enable interoperability in information-rich endeavours, there is currently no unified framework for ontology interoperability itself. Surprisingly little of the state of the art in modularity and structuring, e.g. in software engineering, has been applied to ontology engineering so far. However, application areas like Ambient Assisted Living (AAL), which require synchronisation and orchestration of interoperable services, are in dire need of safe and secure ontology interoperability. OntoIOp (Ontology Integration and Interoperability), a new international standard proposed in ISO/TC 37/SC 3, aims at filling this gap.

1 Introduction and Motivation

In the early 1990s, the widely cited paper "Towards Principles for the Design of Ontologies Used for Knowledge Sharing" by Tom Gruber [13] is credited with a deliberate definition of ontology as a technical term in computer science. Gruber introduced the term to mean a specification of a conceptualisation. That is, an ontology is a description, like a formal specification of a program, of the concepts and relationships that can exist for an agent or a community of agents. This definition is consistent with the usage of ontology as a set of concept definitions, but more general. And it is a different sense of the word than its use in philosophy. On the other hand, as [14] argued, some researchers, drawing inspiration from philosophical ontologies, viewed computational ontology as a kind of applied philosophy.

In the early years of the 20th century, the interdisciplinary project of cognitive science has been bringing the two circles of scholars closer together. For example, there is talk of a "computational turn in philosophy" that includes philosophers analysing the formal ontologies of computer science (sometimes even working directly with the software), while researchers in computer science have been making more references to those philosophers who work on ontology (sometimes with direct consequences for their methods). Still, many scholars in both fields are uninvolved in this trend of cognitive science, and continue to work independently of one another, pursuing separately their different concerns.

Today, ontologies (in the meaning of sets of concept definitions) are applied in eBusiness, eHealth, eGovernment, eInclusion, and virtually all other information-rich endeavours. In addition there are standardisation efforts going on with respect to ontologies in the sense of specifications of a conceptualisation. The multitude of endeavours have created an interoperability problem, which other activities try to overcome—including standardisation efforts. The diversity of current interoperability initiatives demonstrates, however, that there is currently no unified framework within which interoperability efforts themselves can be synchronised and orchestrated. To meet this problem and progress towards safe and secure interoperability standardisation within application areas such as

---

[*] Corresponding author: Oliver Kutz, Research Center on Spatial Cognition (SFB/TR 8), University of Bremen, Cartesium, Enrique-Schmidt Strasse 8, 28359 Bremen, Germany. email: `okutz@informatik.uni-bremen.de`

Ambient Assisted Living (AAL), standards and guidelines are necessary in the area of ontological structuring and modularity. This does not only comprise representing knowledge related to structuring and modularity in an explicit machine-comprehensible way in order to ensure machine-level interoperability, but also annotating and documenting it—not only for the benefit of knowledge engineers and service developers who use an ontology, but also, in a second step, targeting end users of ontology-based services that expose documentation texts from the ontology as online help at runtime. Although in current ontology standardisation initiatives much has already been taken from established software engineering practice, surprisingly little of what is known about modularity and structuring has so far been applied. Part of this gap can be traced back to differing starting points: adopting a description logic basis, as typical within Semantic Web oriented information modelling, has restricted the development of more powerful and generic approaches to supporting modularity. The kernel of the problem is that existing meta-model specifications and ontology definition standards assume that the ontologies produced are essentially compatible down to the exchange of terms and filling in respective knowledge gaps. But this 'assumption' of ontological compatibility frequently fails to hold. It also does not match current practice nor expectations when standardisation is considered across technical communities.

In order to fill the current gaps in modular ontology design and to augment ongoing standardisation efforts with an essential layer of standardised modularity and structuring guidelines, a new working item on "Ontology Integration and Interoperability (OntoIOp)" has been proposed in ISO/TC 37/SC 3 "Systems to manage terminology, knowledge and content". The proposal brings together new results in the international state of the art in ontology-based interoperability. This includes results from several large-scale initiatives. Thus the proposed International Standard OntoIOp is aiming at bridging existing gaps in standards and guidelines.

Pertinent techniques were developed within the European FP7 research project OASIS (Open architecture for Accessible Services Integration and Standardisation) and similar efforts took place in North America and Asia. The synergies targeted within the standardisation activity will, however, allow the resulting International Standard to go significantly beyond the existing efforts and results. In the case of OASIS, for example, the focus is on providing an open platform for interoperable services for the aging; for this the project targets to utilise ICT and other key technologies in order to provide holistic services to older people to support their physical and psychological independence, stimulate their social or psychological engagement and foster their emotional well being. Although relevant for AAL and including AAL components, the focus on services means that the standardisation project will be generic in its approach and applicable for all eApplications beyond the functional and technical requirements of AAL.

While the OWL standard [7] has led to an important unification of notation and semantics, still many diverse formalisms are used for writing ontologies. Some of these, such as RDF, OBO [11] and UML [25], can be seen more or less as fragments and notational variants of OWL, while others, like F-logic [17] and Common Logic [5], clearly go beyond the expressiveness of OWL. Moreover, not only the underlying logics are different, but also the modularity and structuring constructs, and the reasoning methods. Within the European project NeOn, several such languages have been integrated into a unified framework for maintaining networks of ontologies. NeOn also provides a rich tool set for analysis, verification, mapping etc.

Many (domain) ontologies are written in description logics such as $\mathcal{SROIQ}(D)$ (underlying OWL 2 DL) and its fragments. These logics are characterised by having a rather fine-tuned expressivity, exhibiting (still) decidable satisfiability problems, whilst being amenable to highly optimised implementations. However, there are many cases where either weaker DLs are enough—such as sub-Boolean $\mathcal{EL}$ (an OWL 'profile')— and more specialised (and faster) algorithms can be employed, or, contrarily, the expressivity has to be extended beyond the scope of standard DLs.

An example for the former is the NCI thesaurus (containing about 45,000 concepts) which is intended to become the reference terminology for cancer research [24], an example for the latter many foundational ontologies, for instance DOLCE [10], BFO[5], or GFO[6]. Note however that these foundational ontologies also come in different versions ranging in expressivity, typically between OWL and first-order or even second-order logic.

While the web ontology language OWL is being constantly refined and extended, its main target application is the Semantic Web and related areas, and it can thus not be expected to be fit for any purpose: there will always be new, typically interdisciplinary application areas for ontologies where the employed (or required) formal languages do not directly fit into the OWL landscape. *Heterogeneity* (of ontology languages) is thus clearly an important issue. This does not only include cases where the expressivity of OWL is simply exceeded (such as when moving to full first-order logic), but, ultimately, also cases where combinations with or connections to formalism with different semantics have to be covered, such as temporal, spatial, or epistemic logics, cf. e.g. [2, 3, 20, 8, 4].

Ontologies developed, therefore, are often not directly compatible in terms of their syntax and semantics as well as interchange formats, entailing a strong need for interoperability among ontologies. Moreover, the domain knowledge modelled in ontologies is used for providing and integrating software services with hardware devices. Hence, interoperability occurs at various levels:

- interoperability of ontologies
- interoperability of software services, such as web services
- interoperability of hardware devices
- interoperability among ontologies, services and devices

Ontology interoperability is closely related to ontology matching and alignment [9]. Here, we will not so much consider the process of finding suitable matchings and alignments, but rather aim at providing a standardised format for expressing the resulting matchings and alignments. Ontology interoperability is also related to ontology modularity and structuring. Increasingly, different systems have to reuse (different parts of) given ontologies. These (parts of) ontologies have to be agreed upon (i.e. be negotiated between systems) and be systematically linked in order to allow interoperability and the sharing of services. Such (parts of) ontologies can be extracted as modules that are constructed by various module extraction techniques, or reuse previously defined modules that use structuring techniques available in a given ontology language. Ontologies conforming to the structuring guidelines according to a standard provide a significantly improved capability for the goal of 'seamless' data interchange and operation. In addition it is possible to formally guarantee the preservation of particular properties, such as consistency, in specifications so combined.

2 A Sketch of a Future Standard

In the presence of several alternative choices of modelling formalisms, it can be a rather difficult task for an ontology designer to choose an appropriate logic and formalism for a specific ontology design beforehand—and failing in making the right choice might lead to the necessity of re-designing large parts of an ontology from scratch, or limit future expandability. Another issue is the mere size of ontologies making the design process potentially quite hard and error prone (at least for humans), which is particularly a problem for ontologies in the Life Sciences. This issue has been partly cured in OWL by the `imports` construct, but still leaves the problem of 'debugging' large ontologies as an important issue, see e.g. [16]. Also, simple operations such as the re-use of parts of an ontology in a different 'context' whilst *renaming* (parts of) the signature are not possible in the OWL

---

[5] See `http://www.ifomis.org/bfo/`

[6] See `http://www.onto-med.de/ontologies/gfo/`

languages, making it difficult to combine ontologies that use the same terms analysed from different modelling perspectives, thereby easily yielding inconsistencies when performing naive ontology combination.

We here propose a solution to the above issues based on the concept of *heterogeneity*: facing the fact that several logics and formalisms are used for designing ontologies, we suggest heterogeneous structuring constructs that allow to combine ontologies in various ways, in a systematic and formally and semantically well-founded fashion.

The proposed standard will specify a *distributed ontology language* (DOL), which will serve as a language for distributed knowledge representation in ontologies and interoperability among ontologies, and in the long run also among services and devices.

It is important to stress that we do not aim at "yet another ontology language". Rather, DOL will be a meta-language that integrates existing ontology languages and formats. These are included "as-is", such that existing formats and tools can be used without need for conversion. DOL shall include all languages that are currently used for ontology development in practice. Currently, these are the web ontology language OWL together with its sublanguages (called profiles), the uniform modelling language UML (only class diagrams), OBO, Common Logic (and its dialects), F-logic, and others.

DOL will allow the user to integrate and compare ontologies written in different languages. Mappings between ontologies, as they occur in ontology matching and alignment [9], will be first-class citizens. Moreover, the standard will also accommodate for

- interoperability among ontologies, services and devices, as objects definable in the language;
- translations between ontology languages, as objects available in the language;
- heterogeneous ontologies combining parts written in different languages; and
- distributed ontologies in terms of both different internet locations and different ontology languages.

Finally, DOL will feature annotation and documentation facilities—making them available to ontology languages that have not had them so far (such as Common Logic), but also extending the existing facilities of, e.g., OWL, to the new DOL constructs. Section 4 covers this in more detail.

DOL will be defined with formal syntax, namely both a user-readable syntax as well as a distributed ontology interchange format DIF, which is an XML syntax for DOL. Existing ontologies in existing XML-based ontology interchange formats shall be made directly valid ontologies in the distributed ontology interchange format. For non-XML-based ontology interchange formats, such as the OWL Manchester syntax, this should be achieved as well.

DOL will also have a well-defined formal semantics. Since DOL involves several ontology languages, appropriate ontology language translations between the ontology languages integrated into DOL will be defined and standardised. Every pair of ontology languages should have ontology translations into a common target ontology language.

The application T(O) of an ontology language translation T to an ontology O will be part of the DOL syntax. Moreover, there will be projections of DOL to individual ontology languages. Depending on the expressivity of the target ontology language, the projections can be total or only partial. In case of a partial projection, the domain of the projection should be clearly identifiable.

DOL will provide support for the language constructs present in the different module languages (like imports, matchings, renamings, conservative extensions). There should be interoperability among different module languages. DOL will also include one standard module language that can be used as lingua franca. This lingua franca can then be used for modularising ontologies in any supported ontology language. DOL will provide means to make ontology module extraction, which is typically performed by certain tools, explicit, declarative and therefore reproducible independently of the tool. Then e.g. changes in the

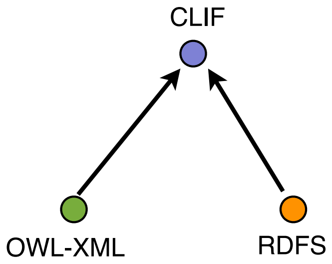

Fig. 1. Mapping two ontology languages into a third one

ontology can be propagated to the extracted module. DOL will also include a syntax for internalising ontology mappings.

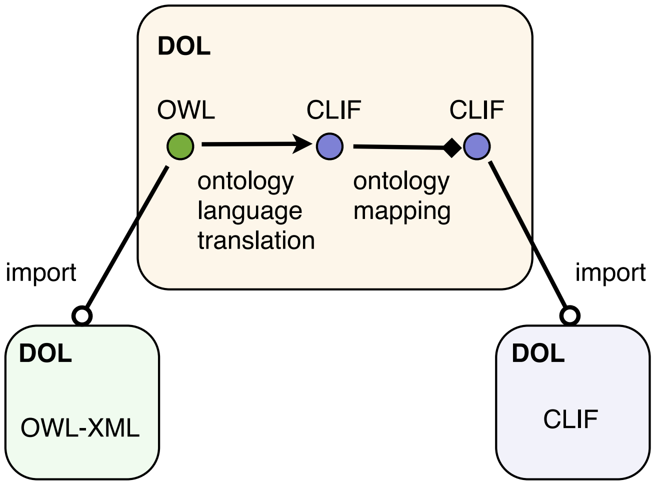

Fig. 2. An ontology mapping between ontologies formulated in different ontology languages

3 A Very Brief Sketch of the Hyperontology Framework

The definition of the semantics of the distributed ontology language DOL leads to technical challenges, since all the individual ontology languages come with their own syntax and semantics, which need to be integrated properly. Addressing this problem of logical translation between ontology languages, [23] has presented the theoretical foundations for the distributed ontology language DOL on the 'translation layer'. The general idea here is that such translations will allow users to use their own preferred ontology formalism, whilst being interoperable with other formalisms.

At the heart of our approach, therefore, is a graph of ontology languages and translations. This graph enables users to:

- relate ontologies that are written in different formalisms;
- re-use ontology modules even if they have been formulated in different formalisms;
- re-use ontology tools like theorem provers and module extractors along translations.

More generally, our approach is based on the theory of institutions (i.e. abstract model theory) and formal structuring techniques from algebraic specification theory. Its main features are the following, paraphrasing [21]:[7]

- The ontology designer can use OBO or OWL to specify most parts of an ontology, and can use first-order (or even higher-order) logic where needed. Moreover, the overall ontology can be assembled from (and can be split up into) semantically meaningful parts ('modules') that are systematically related by structuring mechanisms. These parts can then be re-used and/or extended in different settings.
- Institution theory provides the framework for formalising 'logic translations' between different ontology languages, translating the syntax and semantics of different formalisms. These translations allow in particular the 'borrowing' of reasoning and editing tools from one logic to another, when appropriately related.
- Various concepts of 'ontological module' are covered, including simple imports (extensions) and union of theories, as well as conservative and definitional extensions.
- Structuring into modules is made explicit in the ontology and generates so-called proof obligations for conservativity. Proof obligations can also be used to keep track of desired consequences of an ontology, especially during the design process.
- Re-using (parts of) ontologies whilst renaming (parts of) the signature is handled by *symbol maps* and *hiding symbols*: essentially, this allows the internalisation of (strict) alignment mappings.
- The approach allows heterogeneous refinements: it is possible to prove that an ontology O_2 is a refinement of another ontology O_1, formalised in a different logic. For instance, one can check if a domain ontology is a refinement of (a part of) a foundational one. An interesting by-product of the definition of heterogeneous refinements is that it also provides a rather general definition of heterogeneous sub-ontology and of ontology equivalence.

Tool support for developing heterogeneous ontologies is available via the Heterogeneous Tool Set HETS, which provides parsing, static analysis and proof management for heterogeneous logical theories. HETS visualises the module structure of complex logical theories, using so-called development graphs. For individual nodes (corresponding to logical theories) in such a graph, the concept hierarchy can be displayed. Moreover, HETS is able to prove intended consequences of theories, prove refinements between theories, or demonstrate their consistency. This is done by integrating several first-order provers and model-finders (SPASS, DARWIN), the higher-order prover ISABELLE, as well as DL reasoners like PELLET and FACT++.

A detailed discussion of the various translational relationships between (almost) all known ontology languages can be found in [23]. We here concentrate on the most widely used languages, namely OBO (of specific interest for biomedical ontologies), OWL and its profiles (used for domain ontologies etc.), first- and second-order logic (foundational ontologies), and F-Logic and Common Logic. Fig. 3 illustrates the translational relationships. Leaving out technicalities (but see [23]), a 'regular' translation between two ontology languages means that the syntax and semantics of one logic can be translated into another. This means that, typically, the former is a fragment of the latter. A standard example would be OWL which, via the standard translation, can be considered a fragment of first-order logic. The languages we just mentioned range from sub-Boolean (OWL fragments), sub-first-order (OWL, but comprising full Boolean logic), to full first-order (including a translation from F-logic), to sub-second-order (Common Logic, but extending full first-order by some second-order constructs).

Notice that, inspecting Fig. 3, there are often several translation between two distinct languages, namely one given by an immediate arrow (the 'natural' translation), and one given by composing several other translations. The reason for giving translations that

---

[7] For technical detail and extensive discussion we have to refer to [21].

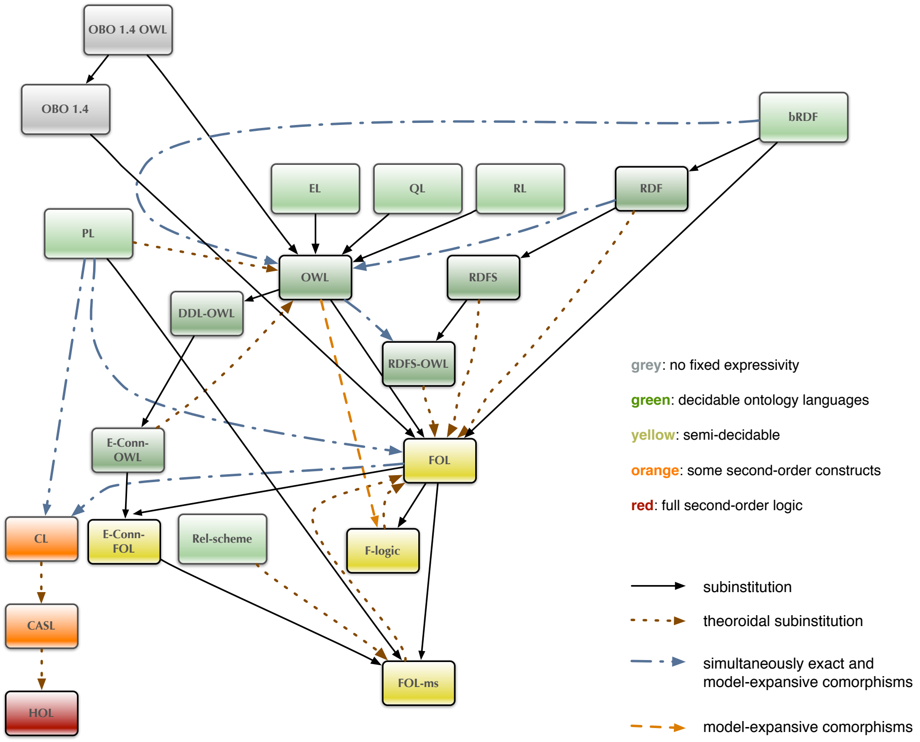

Fig. 3. Translations between ontology languages

could alternatively also be obtained via composition of already existing translations is that the 'natural translations' typically have better properties in terms of comprehensibility and succinctness. The choice of a translation therefore impacts in particular computational properties and tool re-usability.

Note that translations concerning different versions of OBO are of different flavours[8]. The OBO language does not itself come with formal semantics. Beginning with [11], who mapped a fragment of OBO 1.2 to OWL, a semantics for OBO has been assigned by translation. Whilst version 1.3 of OBO, now abandoned, had something similar using Common Logic, the current specification of OBO, version 1.4, gets its semantics entirely via translation to OWL 2. In a sense, thus, the OBO language does not have a fixed logical expressivity, but depends on borrowed model-theoretic semantics from a particular mapping to another ontology language, relative to which corresponding reasoning methods and editing tools can be applied.

Logic translations can in particular be internalised in the ontology languages themselves, in the sense that ontologies can be written in a mix of logical formalisms, where the translations assign respective semantics by operating in the background. For this to work properly, formal structuring principles are necessary, which also need to be discussed when developing the standard.

4 Embedded Documentation

As pointed out initially, ontology documentation addresses not only knowledge engineers and service developers, but potentially even end users. State-of-the-art ontology languages

---

[8] In particular, the progression between the different versions of OBO are only partial, leaving out some language constructs and adding others.

hardly address documentation. Common Logic [5], for example, merely allows for attaching *comments* to its phrases (modules, sentences, imports, or text) and specifies that "a comment is a piece of data". OWL, thanks to its roots in the RDF metadata model, supports subject–property–object annotations. All entities (classes, properties, individuals), axioms, as well as complete OWL ontologies are possible subjects; the supply of possible annotation properties is in principle infinitely extensible (again by ontologies); objects can be multilingual strings, values of other datatypes, or resources identified by URIs. However, no ontology language known to date supports documenting subterms of complex axioms, or subsets of an ontology (which, e.g., have not [yet] been modularised into ontologies of their own)—let alone another approach known from software engineering: literate programming [18], where natural language documentation and formal expressions are freely interwoven and reference each other in a fine-grained way. The particular strength of literate programming is that one can generate both a reference manual and compilable/executable code (here: a formal ontology) from the same source, which facilitates maintenance.

Note that other features planned for DOL (cf. Section 2) also enable better documentation, on a formal level: consider a knowledge model that is formalised as a DL ontology for reasons of decidability or tractability, but suppose that software built on top of that ontology is required to provide further functionality that could only be formalised in first order logic: With a heterogeneous ontology language, such information no longer has to be provided as natural language (as it is common practise) that is opaque to machine assistance. Or consider how an explicit representation of modularity allows for documenting (to knowledge engineers and service developers!) how concepts from other ontologies have been reused.

In previous work, we have provided a proof of concept of an ontology language with rich documentation capabilities [22]. There, we chose OMDoc (Open Mathematical Documents [19]), a rich, foundationally unconstrained semantic markup language for mathematical documents with flexible degrees of formality. We documented, e.g., OWL ontologies by encoding their $\mathcal{SROIQ}(D)$ axioms in OMDoc syntax, using an OMDoc formalisation of the signature of the OWL language as the logical foundation.[9] While this approach guaranteed interoperability in terms of semantics, its *practical* interoperability was limited by the inability to literally reuse fragments of existing ontologies without first translating them to OMDoc and by the incomplete coverage of authoring tools for OMDoc.

Therefore, DOL will make use of the annotation facilities of the existing ontology languages that it comprises, while supporting richer documentation via non-intrusive *standoff markup*, which points to the exact subjects of annotation from external documentation files or from special internal comments. This standoff documentation can be provided as RDF—thus being compatible with the OWL annotation mechanism—or, if fine-grained interweaving is desired, as XML markup with RDF annotations embedded using RDFa [1]. We will enhance the practical usability of this way of annotation by compiling a list of RDF vocabularies/ontologies recommended for annotating ontologies within OntoIOp; this will not only comprise generic metadata vocabularies such as Dublin Core [6], but also vocabularies for advanced ontology engineering demands such as the Ontology Metadata Vocabulary (OMV [15]).

Annotations always have a subject, which is identified by a URI. Where the given ontology language does not provide a way of assigning URIs to a desired subject of an annotation (e.g. if one wants to annotate an import in OWL), DOL will make use of XPointer as a non-destructive means of referencing pieces of text or XML [12]. We intend to utilise the extensibility of the XPointer framework by developing additional XPointer schemes, e.g. for pointing to subterms of Common Logic axioms.

---

[9] Formalising the *axioms* of $\mathcal{SROIQ}(D)$ in OMDoc was not in the focus of this investigation but would also have been possible.

5 Conclusion

We have sketched a new “distributed ontology language”, DOL, integrating various ontology languages, as well as ontology mappings and language mappings, leading to interoperability among ontologies. A formal, machine-comprehensible semantics is provided by the Hyperontology framework and its sophisticated heterogeneous structuring mechanisms, whereas an annotation and documentation framework improves comprehensibility for human users. Future work will have a closer look at the interoperability of ontologies, services and devices as well.

Acknowledgements

Work on this paper has been supported by the EU FP7-funded project “Open architecture for Accessible Services Integration and Standardisation” (OASIS), by the DFG-funded collaborative research centre SFB/TR 8 ‘Spatial Cognition’ and by the German Federal Ministry of Education and Research (Project 01 IW 07002 FormalSafe).

References

1. Adida, B., Birbeck, M., McCarron, S., and Herman, I. RDFa core 1.1. W3C Working Draft, World Wide Web Consortium (W3C).
2. Artale, A., and Franconi, E. A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence 30*, 1-4 (2000), 171–210.
3. Artale, A., Kontchakov, R., Lutz, C., Wolter, F., and Zakharyaschev, M. Temporalising tractable description logics. In *Proc. of the 14th Int. Symposium on Temporal Representation and Reasoning (TIME)* (Washington, DC, USA, 2007), IEEE, pp. 11–22.
4. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. Epistemic first-order queries over description logic knowledge bases. In *Proc. of the 2006 Description Logic Workshop (DL 2006)* (2006), vol. 189 of *CEUR Electronic Workshop Proceedings, http://ceur-ws.org/Vol-189/*.
5. Common Logic Working Group. Common Logic: Abstract syntax and semantics. Tech. rep., 2003.
6. DCMI Usage Board, T. DCMI metadata terms. DCMI recommendation, Dublin Core Metadata Initiative, 2003.
7. Dean, M., and Schreiber, editors, G. OWL Web Ontology Language – Reference. W3C Candidate Recommendation `http://www.w3.org/TR/owl-ref/`, 18 August 2003.
8. Donini, F. M., Lenzerini, M., Nardi, D., Nutt, W., and Schaerf, A. An epistemic operator for description logics. *Artif. Intell. 100*, 1-2 (1998), 225–274.
9. Euzenat, J., and Shvaiko, P. *Ontology Matching.* Springer, Heidelberg, 2007.
10. Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., and Schneider, L. . Sweetening Ontologies with dolce. In *Proc. of EKAW 2002* (2002), vol. 2473 of *LNCS*, Springer, pp. 166–181.
11. Golbreich, C., Horridge, M., Horrocks, I., Motik, B., and Shearer, R. OBO and OWL: Leveraging Semantic Web Technologies for the Life Sciences. In *Proc. of the 6th Int. Semantic Web Conference (ISWC 2007)* (Busan, Korea, November 11-15 2007), K. A. et al., Ed., vol. 4825 of *LNCS*, Springer, pp. 169–182.
12. Grosso, P., Maler, E., Marsh, J., and Walsh, N. W3c xpointer framework. W3C recommendation, World Wide Web Consortium (W3C), Mar. 2003.
13. Gruber, T. R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In *Formal Ontology in Conceptual Analysis and Knowledge Representation* (Deventer, The Netherlands, 1993), N. Guarino and R. Poli, Eds., Kluwer Academic Publishers.
14. Gruber, T. R. Ontology. In *Encyclopedia of Database Systems* (2009), L. Liu and M. T. Özsu, Eds., Springer.
15. Hartmann, J., Palma, R., Sure, Y., Suárez-Figueroa, M. C., Haase, P., Gómez-Pérez, A., and Studer, R. Ontology metadata vocabulary and applications. In *On the Move to Meaningful Internet Systems 2005: OTM Workshops* (Oct. 2005), R. Meersman, Z. Tari, P. Herrero, et al., Eds., no. 3762 in LNCS, Springer, pp. 906–915.

16. Kalyanpur, A., Parsia, B., Horridge, M., and Sirin, E. Finding all Justifications of OWL DL Entailments. In *Proc. of ISWC/ASWC* (2007), vol. 4825 of *LNCS*, Springer, pp. 267–280.
17. Kifer, M., Lausen, G., and Wu, J. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM 42* (July 1995), 741–843.
18. Knuth, D. E. *Literate Programming.* The University of Chicago Press, 1992.
19. Kohlhase, M. OMDoc – *An open markup format for mathematical documents [Version 1.2].* No. 4180 in LNAI. Springer Verlag, Aug. 2006.
20. Kutz, O., Lutz, C., Wolter, F., and Zakharyaschev, M. $\mathcal{E}$-Connections of Abstract Description Systems. *Artificial Intelligence 156*, 1 (2004), 1–73.
21. Kutz, O., Mossakowski, T., and Lücke, D. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis 4*, 2 (2010), 255–333. Special Issue on 'Is Logic Universal?'.
22. Lange, C., and Kohlhase, M. A mathematical approach to ontology authoring and documentation. In *MKM/Calculemus Proceedings* (July 2009), J. Carette, L. Dixon, C. Sacerdoti Coen, and S. M. Watt, Eds., no. 5625 in LNAI, Springer Verlag, pp. 389–404.
23. Mossakowski, T., and Kutz, O. The Onto-Logical Translation Graph. In *Proc. of the 5th Int. Workshop on Modular Ontologies (WoMO-11)* (2011), Frontiers in Artificial Intelligence and Applications, IOS Press.
24. Sioutos, N., de Coronado, S., Haber, M. W., Hartel, F. W., Shaiu, W.-L., and Wright, L. W. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics 40*, 1 (2007), 30–43.
25. Wang, X., and Chan, C. W. Ontology modeling using UML. In *OOIS* (2001), Y. Wang, S. Patel, and R. Johnston, Eds., Springer, p. 59.