Tool Support for Ontology Engineering

Ian Horrocks
Oxford University Computing Laboratory
Oxford, UK
Ian.Horrocks@comlab.ox.ac.uk

No Institute Given

Abstract. The Web Ontology Language (OWL) has been developed and standardised by the World Wide Web Consortium (W3C). It is one of the key technologies underpinning the Semantic Web, but its success has now spread far beyond the Web: it has become the ontology language of choice for a wide range of application domains. One of the key benefits flowing from OWL standardisation has been the development of a huge range of tools and infrastructure that can be used to support the development and deployment of OWL ontologies. These tools are now being used in large scale and commercial ontology development, and are widely recognised as being not simply useful, but essential for the development of the high quality ontologies needed in realistic applications.

1 Introduction

The Web Ontology Language (OWL) [15, 33] has been developed and standardised by the World Wide Web Consortium (W3C). It is one of the key technologies underpinning the Semantic Web, but its success has now spread far beyond the Web: it has become the ontology language of choice for applications in fields as diverse as biology [38], medicine [9], geography [10], astronomy [8], agriculture [40], and defence [26]. Moreover, ontologies are increasingly being used for "semantic data management", and DB technology vendors have already started to augment their existing software with ontological reasoning. For example, Oracle Inc. has recently enhanced its well-known database management system with modules that use ontologies to support 'semantic data management'. Their product brochure[1] lists numerous application areas that can benefit from this technology, including Enterprise Information Integration, Knowledge Mining, Finance, Compliance Management and Life Science Research.

The standardisation of OWL has brought with it many benefits. In the first place, OWL's basis in description logic has made it possible to exploit the results of more than twenty-five years of research and to directly transfer theoretical results and technologies to OWL. As a consequence, the formal properties of OWL entailment are well understood: it is known to be decidable, but to have

[1] http://www.oracle.com/technology/tech/semantic_technologies/pdf/oracle%20db%20semantics%20overview%2020080722.pdf

high complexity (NExpTime-complete for OWL and 2NExpTime-complete for OWL 2 [34]). Moreover, algorithms for reasoning in OWL have been published, and implemented reasoning systems are widely available [30, 44, 12, 39]. These systems are highly optimised and have proven to be effective in practice in spite of the high worst-case complexity of standard reasoning tasks.

One of the key benefits flowing from OWL standardisation has been the subsequent development of a huge range of tools and infrastructure that can be used to support the development and deployment of OWL ontologies. These include editors and ontology development environments such as Protégé-OWL,[2] TopBraid Composer[3] and Neon;[4] reasoning systems such as HermiT [30], FaCT++ [44], Pellet [39], and Racer [12]; explanation and justification tools such as the Protégé OWL Debugger[5] and the OWL explanation workbench [14]; ontology mapping and integration tools such as Prompt [32] and ContentMap [32, 23]; extraction and modularisation tools such ProSÉ;[6] comparison tools such as OWLDiff;[7] and version control tools such as ContentCVS [22].

Such tools are now being used in large scale and commercial ontology development, and are widely recognised as being not simply useful, but essential for the development of the high quality ontologies needed in realistic applications.

2 Background

Ontologies are formal vocabularies of terms, often shared by a community of users. One of the most commonly used ontology modelling languages is the Web Ontology Language (OWL), which has been standardised by the World Wide Web Consortium (W3C) [15]; the latest version, OWL 2, was released in October 2009 [33]. OWL's formal underpinning is provided by description logics (DLs) [2]—knowledge representation formalisms with well-understood formal properties.

A DL ontology typically consists of a *TBox*, which describes general relationships in a domain, and an *ABox*, which describes information about particular objects in the domain. In a comparison with relational databases, a TBox is analogous to a database schema, and an ABox is analogous to a database instance; however, DL ontology languages are typically much more expressive than database schema languages.

Many ontology-based applications depend on various *reasoning tasks*, such as *ontology classification* and *query answering*, which can be solved using *reasoning algorithms*. Two types of reasoning algorithms for DLs are commonly used. *Tableau algorithms* [4, 16, 18, 17] can be seen as model-building algorithms: to

[2] `http://protege.stanford.edu/overview/protege-owl.html`
[3] `http://www.topbraidcomposer.com/`
[4] `http://neon-toolkit.org/`
[5] `http://www.co-ode.org/downloads/owldebugger/`
[6] `http://krono.act.uji.es/people/Ernesto/safety-ontology-reuse`
[7] `http://krizik.felk.cvut.cz/km/owldiff/`

show that a DL ontology $\mathcal{K}$ does not entail a conclusion, these algorithms construct a model of $\mathcal{K}$ that invalidates the conclusion. In contrast, *resolution-based algorithms* [31, 20, 19, 21] show that $\mathcal{K}$ entails a conclusion by demonstrating that $\mathcal{K}$ and the negation of the conclusion are contradictory. *Reasoners* are software components that provide reasoning services to other applications. Reasoners such as Pellet [35], FaCT++ [45], RACER [11], CEL [3], and KAON2 [29] provide reasoning services for a range of DLs, and have been used in many applications.

Medicine and the life sciences have been prominent early adopters of ontologies and ontology based technologies, and there are many high profile applications in this area. For example, the Systematised Nomenclature of Medicine — Clinical Terms (SNOMED CT) [42] is a clinical ontology being developed by the International Health Terminology Standards Development Organisation (IHTSDO),[8] and used in healthcare systems of more than 15 countries, including Australia, Canada, Denmark, Spain, Sweden and the UK. GALEN [41] is a similar open-source ontology that has been developed in the EU-funded FP III project GALEN and the FP IV framework GALEN-In-Use.[9] The Foundational Model of Anatomy (FMA) [37] is an open-source ontology about human anatomy developed at the University of Washington. The National Cancer Institute (NCI) Thesaurus [13] is an ontology that models cancer diseases and treatments. The OBO Foundry[10] is a repository containing about 80 biomedical ontologies developed by a large community of domain experts.

Ontologies such as SNOMED CT, GALEN, and FMA are gradually superseding the existing medical classifications and will provide the future platforms for gathering and sharing medical knowledge; in the UK, for example, SNOMED CT is being used in the National Programme for Information Technology (NPfIT) being delivered by "NHS Connecting for Health".[11] Capturing medical records using ontologies will reduce the possibility for data misinterpretation, and will enable information exchange between different applications and institutions, such as hospitals, laboratories, and government statistical agencies. Apart from providing a taxonomy of concepts/codes for different medical conditions, medical ontologies such as SNOMED CT, GALEN, and FMA describe the precise relationships between different concepts. These ontologies are extensible *at point of use*, thus allowing for "post-coordination": users can add new terms (e.g., "almond allergy"), which are then seamlessly integrated with the existing terms (e.g., as a subtype of "nut allergy"). Clearly, the correctness of such (extended) ontologies is of great importance, as errors could adversely impact patient care.

Medical ontologies are strongly related to DLs and ontology languages. In fact, SNOMED CT can be expressed in the description logic $\mathcal{EL}$++ [1], a well known sub-boolean DL that is the basis for the EL profile of OWL 2 [34]. GALEN, although originally developed using the GRAIL description logic lan-

[8] `http://www.ihtsdo.org/`
[9] `http://www.opengalen.org/`
[10] `http://www.obofoundry.org/`
[11] `http://www.connectingforhealth.nhs.uk/`

guage [36], has now been translated into OWL.[12] FMA was not originally modelled using description logics, but has also been translated into OWL [9]. The developers of medical ontologies have recognised the numerous benefits of using a DL based ontology language, such as the unambiguous semantics for different modelling constructs, the well-understood tradeoffs between expressivity and computational complexity [2, Chapter 3], and the availability of provably correct reasoners and tools.

The development and application of medical ontologies such as SNOMED CT, GALEN, and FMA crucially depend on various reasoning tasks. Ontology classification (i.e., organising classes into a specialisation/generalisation hierarchy) plays a major role during ontology development, as it provides for the detection of potential modelling errors [46]. For example, about 180 missing subclass relationships were detected when the version of SNOMED CT used by the NHS was classified using FaCT++ [47]. Furthermore, ontology classification can aid users in merging different ontologies [28], and it allows for ontology validation [5, 7]. In contrast, query answering is mainly used during ontology-based information retrieval [43]; e.g., in clinical applications query answering might be used to retrieve "all patients that suffer from nut allergies".

The benefits of reasoning enabled tools for supporting ontology engineering are now recognised well beyond the academic setting. For example, OWL reasoning tools are currently being used by British Telecom in the above mentioned NPfIT project, and other companies involved in this project, such as Siemens, are actively applying DLs as a conceptual modelling language.

3 Reasoning Support for Ontology Engineering

SNOMED CT is extremely large: it currently defines approximately 400,000 classes. Developing ontologies, in particular such large ontologies, is extremely challenging. Large and often distributed teams of domain experts may develop and maintain the ontology over the course of many years. It is useful if not essential to support such development and maintenance processes with sophisticated tools that help users to identify possible errors in their formalisation of domain knowledge.

For example, a reasoner can be used to identify inconsistent classes; that is, classes whose extension is necessarily empty. This typically indicates an error in the ontology as it is unlikely that the knowledge engineer intended to introduce a class that can have no instances and that is, in effect, simply a synonym for the built-in OWL Nothing class (the inconsistent class). Similarly, the reasoner can be used to recognise when two different classes are semantically equivalent; that is, classes whose extensions must always be the same. This may indicate an error or redundancy in the ontology, although it is also possible that multiple names for the same concept have deliberately been introduced—e.g., Heart-Attack and Myocardial-Infarction.

[12] `http://www.co-ode.org/galen/`

We can think of inconsistent classes as being over-constrained. A much more typical error in practice is that classes are under-constrained; this can arise because important facts about the class may be so obvious to human experts that they forget to explicate them or simply assume that they must hold. One very common example is missing disjointness assertions: a human expert may, for example, simply assume that concepts such as Arm and Leg are disjoint. A reasoner can also be used to help identify this kind of error—the reasoner is used to compute a hierarchy of classes based on the sub-class relationship, and this computed hierarchy can then be examined by human experts and compared to their intuition about the correct hierarchical structure of domain concepts.

This is not just a theory; reasoning enabled ontology tools have by now proved themselves in realistic applications. For example, an OWL tool was used at the Columbia Presbyterian medical centre in order to correct important errors in the ontology used for classifying pathology lab test results; if these errors had gone uncorrected, then they could have had a serious and adverse impact on patient care [25]. Similarly, Kaiser Permanente,[13] a large health care provider in the USA, is using the Protégé-OWL ontology engineering environment and the HermiT OWL 2 reasoner to develop an extended and enriched version of SNOMED-CT; in the following section we will examine this project in more detail and see how reasoning is being used to support the development of the extended ontology.

3.1 Extending SNOMED CT

In order to support a wider range of intelligent applications, Kaiser Permanente need to extend SNOMET CT in a number of different directions. In the first place, they need to express concepts whose definition involves negative information. For example, they need to express concepts such as: Non-Viral-Pneumonia; that is, a Pneumonia that is *not* caused by a Virus. In the second place, they need to express concepts whose definition involves disjunctive information. For example, they need to express concepts such as Infectious-Pneumonia; that is, a Pneumonia that is caused by a Virus *or* a Bacterium. Finally, they need to express concepts whose definition includes cardinality constraints. For example they need to express concepts such as Double-Pneumonia; that is, a Pneumonia that occurs in *two* Lungs.

Such concepts can be relatively easily added to the OWL version of SNOMED CT using a tool such as Protégé-OWL, and the reasoning support built in to Protégé-OWL can be used to check if the extended ontology contains inconsistent classes, or entailments that do not correspond to those expected by domain experts. After performing various extensions, including those mentioned above, all ontology classes were found to be consistent. However, the reasoner failed to find expected subsumption entailments; for example, the ontology does *not* entail that Bacterial-Pneumonia is a kind of Non-Viral-Pneumonia. This entailment was

[13] `https://www.kaiserpermanente.org/`

expected by domain experts because a pneumonia that is caused by a bacterium is not caused by a virus.

The reason for this missing entailment is that the SNOMED CT ontology is highly under-constrained. For example, it does not explicitly assert "intuitively obvious" class disjointness; in particular, it does not assert that Virus and Bacterium are disjoint. Having identified this problem, the needed disjointness axioms were added to the extended version of SNOMED CT.

After adding these axioms, many additional desired subsumptions were entailed, including the one between Bacterial-Pneumonia and Non-Viral-Pneumonia. Unfortunately, the OWL reasoner also revealed that previously consistent classes had become inconsistent in the extended ontology; one example of such a class was Percutanious-Embolization-of-Hepatic-Artery-Using-Fluoroscopy-Guidance. By using explanation tools, it was discovered that the reason for these inconsistencies were SNOMED CT classes such as Groin that describe "junction" regions of anatomy—in the case of Goin, the junction between the Abdomen and the Leg. In SNOMED CT, these junction regions are defined using simple subsumption axioms; for example, Groin is defined as a subclass of Abdomen and a subclass of Leg. When Abdomen and Leg are asserted to be disjoint, as is obviously intended, any instance of Groin would thus need to be an instance of two disjoint classes, and Groin is thus inconsistent. This reveals a serious modelling error in SNOMED CT—modelling such junction regions in this way is simply not correct.

Correct modelling of (concepts such as) Groin turns out to be quite complex. After considerable effort, it was determined that an appropriate axiomatisation would be something like the following:

$$\begin{aligned}
\mathsf{Groin} &\sqsubseteq \exists \mathsf{hasPart}.(\exists\ \mathsf{isPartOf}.\mathsf{Abdomen}))\\
\mathsf{Groin} &\sqsubseteq \exists \mathsf{hasPart}.(\exists \mathsf{isPartOf}.\mathsf{Leg})\\
\mathsf{hasPart} &\equiv \mathsf{isPartOf}^{-}\\
\mathsf{Groin} &\sqsubseteq \forall \mathsf{hasPart}.(\exists \mathsf{isPartOf}.(\mathsf{Abdomen} \sqcup \mathsf{Leg}))
\end{aligned}$$

In this axiomatisation it is stated that the groin consists of two parts, one of which is part of the abdomen and one of which is part of the leg. The axiomatisation introduces the use of inverse roles as well as universal quantification, suggesting that quite an expressive ontology language is needed for precise modelling of anatomical terms.

As well as illustrating the importance of reasoning enhanced ontology engineering tools, extending SNOMED CT in this way also revealed the importance of explanation. In particular, the inconsistencies that arose after the addition of the disjointness axioms were very difficult for domain experts to understand, to the extent that the correctness of these entailments was initially doubted. If explanation tools had not been available, the experts would very likely have lost faith in the reasoning tools, and probably would have stopped using them. By using explanation systems they were able to understand the cause of the problem, to see that the initial design of the ontology was faulty, and to devise a more appropriate axiomatisation.

4 Other Tools

In addition to ontology engineering environments, a large range of other tools is now becoming available. This includes, for example, tools supporting ontology integration and modularisation, ontology comparison, and ontology version control.

When developing a large ontology, it is useful if not essential to divide the ontology into modules in order to make it easier to understand and to facilitate parallel work by a team of ontology engineers. Similarly, it may be desirable to extract from a large ontology a module containing all the information relevant to some subset of the domain—the resulting small(er) ontology will be easier for humans to understand and easier for applications to use. New reasoning services can be used both to alert developers to unanticipated and/or undesirable interactions when modules are integrated, and to identify a subset of the original ontology that is indistinguishable from it when used to reason about the relevant subset of the domain [6].

These techniques have been implemented in tools such as ProSÉ.[14] Given a subset of the vocabulary used in an ontology, ProSÉ can be used to extract a module that includes all the axioms relevant to that vocabulary. The extraction technique uses the semantics of the ontology rather than its syntax, and is based on the logical notion of conservative extensions [27]. It is this formal basis that allows the very strong semantic guarantee to be provided, i.e., the guarantee that, for any entailment question that uses concepts formed only from the given vocabulary, the answer computed using the module will be the same as that computed using the original ontology.

ContentMap[15] is an example of a tool that uses the same underlying semantic framework to support ontology integration. It uses semantic techniques to compute new entailments that would arise as a result of merging a pair of ontologies using a given set of integration axioms (axioms using terms from both ontologies). Users can say whether or not these new entailments are desired, and the tool suggests "repair plans"; these are minimal sets of changes that invalidate undesired entailments while retaining desired ones [24].

ContentCVS[16] is an example of a tool that supports ontology versioning. It uses the well known CVS paradigm, adapting it to the case of ontologies by using a combination of syntactic and semantic techniques to compare ontology versions [22].

5 Discussion

As we have seen in Section 3.1, reasoning enabled tools provide vital support for ontology engineering. Ontology development environments such as Protégé-OWL

[14] `http://krono.act.uji.es/people/Ernesto/safety-ontology-reuse/proSE-current-version`

[15] `http://krono.act.uji.es/people/Ernesto/contentmap`

[16] `http://krono.act.uji.es/people/Ernesto/contentcvs`

are now considered a minimum requirement for serious ontology engineering tasks, and a wide range of additional tools and infrastructure is now becoming available.

Experience with these tools has illustrated some of the complexities of ontology development, and suggests a high likelihood that non-trivial ontologies developed without tool support will contain errors. These may be errors of omission, typically where the ontology engineer(s) forget to add "obvious" information to the ontology; they may also be errors of commission, where concepts have been over-constrained or incorrectly modelled. Re-use and/or modular ontology design also introduces the possibility that, while individually correct, merging ontologies can reveal incompatibilities in their design.

It has also become evident that, as well as identifying the existence of errors, it is essential for tools to be able to pinpoint errors, explain the reasoning involved in the unexpected (non-) entailment, and if possible offer repair suggestions. Without this facility, domain experts may be unable to identify the source of errors; this may even cause them to lose faith in the correctness of the reasoning system, and ultimately to stop using it.

References

1. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ Envelope. In *Proc. IJCAI-05*, pages 364–369, Edinburgh, UK, 2005.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook.* Cambridge University Press, 2003.
3. F. Baader, C. Lutz, and B. Suntisrivaraporn. CEL—A Poly-nomial-time Reasoner for Life Science Ontologies. In *Proc. IJCAR'06*, pages 287–291, Seattle, WA, USA, 2006.
4. F. Baader and U. Sattler. An Overview of Tableau Algorithms for Description Logics. *Studia Logica*, 69:5–40, 2001.
5. O. Bodenreider, B. Smith, A. Kumar, and A. Burgun. Investigating subsumption in SNOMED CT: An exploration into large description logic-based biomedical terminologies. *Artificial Intelligence in Medicine*, 39(3):183–195, 2007.
6. B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular reuse of ontologies: Theory and practice. *J. of Artificial Intelligence Research*, 31:273–318, 2008.
7. O. Cure and J. Giroud. Ontology-based data quality enhancement for drug databases. In *Proc. of Int. Workshop on Health Care and Life Sciences Data Integration for the Semantic Web*, 2007.
8. S. Derriere, A. Richard, and A. Preite-Martinez. An ontology of astronomical object types for the virtual observatory. *Proc. of Special Session 3 of the 26th meeting of the IAU: Virtual Observatory in Action: New Science, New Technology, and Next Generation Facilities*, 2006.
9. C. Golbreich, S. Zhang, and O. Bodenreider. The foundational model of anatomy in OWL: Experience and perspectives. *J. of Web Semantics*, 4(3):181–195, 2006.
10. J. Goodwin. Experiences of using OWL at the ordnance survey. In *Proc. of the First Int. Workshop on OWL Experiences and Directions (OWLED 2005)*, volume 188 of *CEUR Workshop Proceedings*. CEUR (`http://ceur-ws.org/`), 2005.

11. V. Haarslev and R. Möller. RACER System Description. In *Proc. IJCAR 2001*, pages 701–706, Siena, Italy, 2001.
12. V. Haarslev, R. Möller, and M. Wessel. Querying the Semantic Web with Racer + nRQL. In *Proc. of the KI-2004 Intl. Workshop on Applications of Description Logics (ADL'04)*, 2004.
13. F. W. Hartel, S. de Coronado, R. Dionne, G. Fragoso, and J. Golbeck. Modeling a description logic vocabulary for cancer research. *J. of Biomedical Informatics*, 38(2):114–129, 2005.
14. 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 *Lecture Notes in Computer Science*, pages 323–338. Springer, 2008.
15. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.
16. I. Horrocks and U. Sattler. A tableaux decision procedure for $\mathcal{SHOIQ}$. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 448–453, 2005.
17. I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
18. I. Horrocks, U. Sattler, and S. Tobies. Reasoning with Individuals for the Description Logic $\mathcal{SHIQ}$. In *Proc. CADE-17*, pages 482–496, Pittsburgh, PA, USA, 2000.
19. U. Hustadt, B. Motik, and U. Sattler. Reasoning in Description Logics with a Concrete Domain in the Framework of Resolution. In R. L. de Mántaras and L. Saitta, editors, *Proc. of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, pages 353–357, Valencia, Spain, August 22–27 2004. IOS Press.
20. U. Hustadt, B. Motik, and U. Sattler. Reducing $\mathcal{SHIQ}^{-}$ Description Logic to Disjunctive Datalog Programs. In D. Dubois, C. A. Welty, and M.-A. Williams, editors, *Proc. of the 9th Int. Conference on Principles of Knowledge Representation and Reasoning (KR 2004)*, pages 152–162, Whistler, Canada, June 2–5 2004. AAAI Press.
21. U. Hustadt, B. Motik, and U. Sattler. A Decomposition Rule for Decision Procedures by Resolution-based Calculi. In F. Baader and A. Voronkov, editors, *Proc. of the 11th Int. Conference on Logic for Programming Artificial Intelligence and Reasoning (LPAR 2004)*, volume 3452 of *LNAI*, pages 21–35, Montevideo, Uruguay, March 14–18 2005. Springer.
22. E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga Llavori. ContentCVS: A CVS-based Collaborative ONTology ENgineering Tool (demo). In *Proc. of the 2nd Int. Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2009)*, volume 559 of *CEUR* (`http://ceur-ws.org/`), 2009.
23. E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga Llavori. Logic-based ontology integration using ContentMap. In A. Vallecillo and G. Sagardui, editors, *Proc. of XIV Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2009)*, pages 316–319, 2009.
24. E. Jiménez-Ruiz, B. Cuenca Grau, I. Horrocks, and R. Berlanga Llavori. Ontology integration using mappings: Towards getting the right logical consequences. In *Proc. of the 6th European Semantic Web Conf. (ESWC 2009)*, volume 5554 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2009.
25. A. Kershenbaum, A. Fokoue, C. Patel, C. Welty, E. Schonberg, J. Cimino, L. Ma, K. Srinivas, R. Schloss, and J. W. Murdock. A view of OWL from the field:

Use cases and experiences. In *Proc. of the Second Int. Workshop on OWL Experiences and Directions (OWLED 2006)*, volume 216 of *CEUR Workshop Proceedings*. CEUR (`http://ceur-ws.org/`), 2006.

26. L. Lacy, G. Aviles, K. Fraser, W. Gerber, A. Mulvehill, and R. Gaskill. Experiences using OWL in military applications. In *Proc. of the First Int. Workshop on OWL Experiences and Directions (OWLED 2005)*, volume 188 of *CEUR Workshop Proceedings*. CEUR (`http://ceur-ws.org/`), 2005.
27. C. Lutz, D. Walther, and F. Wolter. Conservative extensions in expressive description logics. In *Proc. of the 20th Int. Joint Conf. on Artificial Intelligence (IJCAI 2007)*, pages 453–458, 2007.
28. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In *Proc. KR 2000*, pages 483–493, Breckenridge, CO, USA, 2000.
29. B. Motik and U. Sattler. A Comparison of Reasoning Techniques for Querying Large Description Logic ABoxes. In *Proc. LPAR 2006*, pages 227–241, 2006.
30. B. Motik, R. Shearer, and I. Horrocks. Hypertableau reasoning for description logics. *J. of Artificial Intelligence Research*, 36:165–228, 2009.
31. H. D. Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-Based Methods for Modal Logics. *Logic Journal of the IGPL*, 8(3):265–292, 2000.
32. N. F. Noy and M. A. Musen. The PROMPT suite: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
33. OWL 2 Web Ontology Language Overview. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-overview/`.
34. OWL 2 Web Ontology Language Profiles. W3C Recommendation, 27 October 2009. Available at `http://www.w3.org/TR/owl2-profiles/`.
35. B. Parsia and E. Sirin. Pellet: An OWL-DL Reasoner. Poster, In Proc. ISWC 2004, Hiroshima, Japan, 2004.
36. A. L. Rector, S. Bechhofer, C. A. Goble, I. Horrocks, W. A. Nowlan, and W. D. Solomon. The GRAIL concept modelling language for medical terminology. *Artificial Intelligence in Medicine*, 9(2):139–171, 1997.
37. C. Rosse and J. V. L. Mejino. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics*, 36:478–500, 2003.
38. A. Sidhu, T. Dillon, E. Chang, and B. S. Sidhu. Protein ontology development using OWL. In *Proc. of the First Int. Workshop on OWL Experiences and Directions (OWLED 2005)*, volume 188 of *CEUR Workshop Proceedings*. CEUR (`http://ceur-ws.org/`), 2005.
39. E. Sirin, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *J. of Web Semantics*, 5(2):51–53, 2007.
40. D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: The AGROVOC example. *J. of Digital Information*, 4(4), 2004.
41. W. Solomon, A. Roberts, J. E. Rogers, C. J. W. C.J., and A. L. Rector. Having our cake and eating it too: How the GALEN Intermediate Representation reconciles internal complexity with users' requirements for appropriateness and simplicity. In *Proc. AMIA 2000*, pages 819–823, CA, USA, 2000.
42. K. A. Spackman. SNOMED RT and SNOMEDCT. Promise of an international clinical terminology. *M.D. Computing*, 17(6):29, 2000.

43. R. Stevens, P. G. Baker, S. Bechhofer, G. Ng, A. Jacoby, N. W. Paton, C. A. Goble, and A. Brass. TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources. *Bioinformatics*, 16(2):184–186, 2000.
44. D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. Springer, 2006.
45. D. Tsarkov and I. Horrocks. FaCT++ Description Logic Reasoner: System Description. In *Proc. IJCAR 2006*, pages 292–297, Seattle, WA, USA, 2006.
46. K. Wolstencroft, R. McEntire, R. Stevens, L. Tabernero, and A. Brass. Constructing ontology-driven protein family data-bases. *Bioinformatics*, 21(8):1685–1692, 2005.
47. C. Wroe. Is Semantic Web technology ready for Healthcare? In *Proc. of ESWC'06 Industry Forum*, volume 194 of *CEUR (`http://ceur-ws.org/`)*, 2006.