Barriers to the use of OWL in Knowledge Driven Applications

Alan Rector & Robert Stevens

School of Computer Science, University of Manchester
rector@cs.manchester.ac.uk

Abstract: Using OWL for commercial and other widely deployed knowledge driven applications raises major issues from the point of software engineering: a) Predictability – will the software be reliable and can it be debugged quickly when errors occur? b) Usability – can users see what they require and is what they need "handy"? c) The need for rich metadata – many ontologies exist for their annotations, to treat them as incidental makes it unlikely that users needs will be satisfied. Although some solutions require long term research, strategies to mitigate the problems are possible now and urgently needed.

1. Introduction

The authors are developing industrial strength applications using ontologies implemented in OWL to drive information systems. Our primary experience is in biomedical applications, but we doubt that the issues raised here are peculiar to biomedicine. Some of what is proposed here is already under development or available in some systems. However, we contend without higher priority and standard solutions, OWL is unlikely to achieve wide use, at least for the types of knowledge driven applications that we are developing.

For background, our goal is to build (relatively) generic applications that derive their content and specific behaviours from knowledge bases implemented in OWL generalizing the work in [1]. These applications have several features in common:

- They have large T-Boxes, often tens or even hundreds of thousands of classes. They may, or may not, have large A-Boxes.
- They must represent both the domain knowledge – diseases, genes, anatomy, etc – and the data and data structures that hold the corresponding information about individuals – medical records, research protocols and data stores, etc.
- The knowledge held in the systems must be, in large part, developed by users, or at least configuration engineers with minimal knowledge of OWL but extensive experience of other systems and expectations based on that experience.
- They are designed to be both standard and highly tailorable within that standard – in fact tailorability is a prime reason for adopting knowledge based solutions.
- They require large amounts of meta-data and annotation – language, mappings, procedural attachments, indexing of external resources, etc.

Examples of systems based on related knowledge representations date back at least twenty years (*e.g.* OPAL [2], Pen&Pad [3]) and related applications are being built

using rival technologies such as XML-Schema (*e.g.* Pedro [4]). The ability to compose definitions of new concepts out of old, infer equivalence and classification are strong motivations for using OWL. However, users tend to be prepared only to take on new capabilities if their existing requirements are met. At present, OWL fails to match their current systems in mission critical ways.

2. Key issues: Predictability, Usability, and Metadata

2.1 Predictability, Resource constrained reasoning, and Coping with inconsistent A-Boxes.

OWL reasoners are brittle. They can fail unexpectedly and provide minimal feedback in at least two situations:

- Small changes to an OWL ontology can causes the time required for classification to explode exponentially.
- An error causing an A-Box to be inconsistent.

In both cases, most current reasoners provide minimal information and minimal control. Most users would rather receive a partial answer in predictable time than a complete answer in unpredictable time. We suggest that three approaches to mitigation be treated as urgent:

- Resource constrained reasoning, whereby the user can determine an upper limit for the time to be spent on any subproblem and receive a comprehensible report on which subproblems exceed this limit. Constrained reasoning is already supported by Racer but not by any freely available reasoners.
- Improved logging so that the source of unexpectedly slow reasoner can be located, as is already supported to some degree by Pellet. [5]
- Separation of reasoning over the T-Box from reasoning over the A-Box plus a further check that a "T-box approximation" of the A-Box be shown not to contain unsatisfiable classes before attempting to classify the A-Box. By a "T-Box" approximation, we mean a T-Box in which all individuals are replaced by classes, all type axioms by subclass axioms, and all facts by existential restrictions.

2.2 Usability: Local inspectability and "Sanctioning" / "Slot attachment"

OWL ontologies are hard to understand. Understanding is made harder because:

- OWL ontologies depend on classification, but most reasoners can only classify an ontology *de novo*, forcing development environments to treat them as "compilers", invoked only occasionally. The result is that authors of large ontologies see the results of their work only sporadically.
- OWL axioms have global consequences, so that an ontology author rarely has all the information they need locally visible and "to hand." The interpretation of domain and range declarations as universal restrictions is particularly problematic.
- Other systems, such as XML-Schema, Frames, or RDF(S) provide a "template" for what can be said. OWL provides only restrictions as to what may not be said. Even restrictions on what may not be said depend on the ontology including all

relevant disjoint statements and universal restrictions which are difficult, and often counter-productive, to make in advance. Therefore, during most of development, authors must browse exhaustively or search lexically. Users often seek work arounds to this problem, *e.g.* inserting universal constraints (see). These often result in statements that are patently false according to the first-order semantics.

We suggest three measures to address these issues:

- Incremental classification and rapid saving and reloading of the current state of reasoners, so that authors see work against the a check-pointed inferred ontology, make changes, see and then incremental effects immediately. (Racer, and reportedly the latest version of Pellet support at least some of this functionality.)
- The option of an alternative treatment of domain and range declarations, to limit unexpected global effects. In many uses, any inference from a domain or range restriction is an error. Therefore, it would often be better to treat them "constraints" as described by Motik and Sattler [6], or alternatively, to perform classification first ignoring domain and range constraints, which then be checked.
- The option to use a template mechanism based to provide the information on "what can be said" similar to "slot attachment" in frames. This would avoid users inserting (frequently incorrect) first-order statements as "work arounds" to achieve their required behaviour.

2.3 Rich metadata and Multi-layer systems

Many ontologies – witness GO and OBOEdit – exist primarily to index their metadata. The relegation of metadata to the status of mere "annotations" reflects their marginal status. This means the language does not standardize the expression of much information essential to user applications.

- Standard sets of annotation properties, with specification of their value ranges and the domain of entities to which they apply, and how they fit into hierarchies of values are as important to many applications as standard sets of domain entities.
- Many ontology tools must depend on meta-data to determine how different parts of the ontology are to be presented to users – *e.g.* whether they should be "visible" as proper domain knowledge or "internal" to the workings of the ontology.
- Lexical information is often much richer than just a simple label in a language.
- Editorial and provenance information is as important for ontologies as for any other software or knowledge artifact, and the editorial and provenance information itself often requires editorial and provenance information.
- Many applications must represent both the domain knowledge and the data structures and user interface objects that hold that domain knowledge. The two must be kept distinct. Patients do not have missing pulse rates; data structures do. Layered versions of OWL and description logics have been proposed [7] but have so far made little impact.
- Many applications require references to classes themselves rather than to all members of each class. A request for a book on "vertebrate anatomy" is not met by a book on "human anatomy" let alone a book on the "anatomy of the Elephant Man," even though both satisfy the restriction "about some Vertebrate". (The controversy and lack of standard approach to this issue is documented in [8])

The entire issue of metadata, annotation, provenance, and multi-layer systems requires a thorough review beyond the scope of a brief paper. However, several steps could be taken now.

- Drive forward the Rich Annotation proposals in the OWL 2 committee, including defining "projections," as a systematic basis for a migration path to richer semantics and develop into a standard means for layered modelling.
- Strongly support the use of the richer SKOS annotations "skos:preflabel", "skos:altlabel", "skos:hiddenlabel" and collaborate with other W3 working groups on richer lexical representations.
- Provide the notions of domain, range, and subproperties for annotation properties to indicate where and how to use annotations and how they are grouped.
- Provide an agreed standard for referring to classes as values or "subjects", *e.g.* an agreed prefix, so that there is an agreed way to refer to subjects in OWL ontologies – *e.g.* to "a book on vertebrate anatomy".

3. Conclusion

OWL provides a strong logical basis for ontologies. However, to address real use cases for knowledge driven applications, it must also address a wider set of issues related to ontology engineering, ease of use, and reliability. It must also address the fundamental fact that not all information pertinent to applications can be captured in a single first-order theory. Users neither will nor can migrate from their existing methods to take advantage of OWL unless it addresses their existing requirements as well as opening new possibilities.

Acknowledgements

This work supported in part by the JISC and UK EPSRC projects CO-ODE and HyOntUse (GR/S44686/1), the EU funded Semantic Mining Network of Excellence.

References

1. Pulestin C, Parsia B, Cunnngham J, Rector A; Building hybrid ontology-backed software models. 2008; Int Conference on Semantic Web (ISWC-2008): Karlsruhe, De: (in press).
2. Musen MA, Fagan LM, Combs DM, Shortliffe EH. Use of a domain model to drive an interactive knowledge-editing tool. *In Journal of Man-Machine Studies*. 1987;26:105-121.
3. Nowlan WA. Clinical workstation: Identifying clinical requirements and understanding clinical information. *International Journal of Bio-Medical Computing*. 1994;34:85-94.
4. Jameson D, Garwood K, Garwood C, et al. Data capture in bioinformatics:requirements and experiemences with Pedro. *BMC Bioinformaitcs*. 2008;9:183.
5. Wang TD, Parsia B; Ontology performance profiling and model examination: First steps. 2007; International Semantic Web Conference (ISWC 2007): 595-608.
6. Motik B, Horrocks I, Sattler U; Adding integrity constraints to OWL. 2007; Third OWL Experiences and Directions Workshop (OWLEd-2007): CEUR #258;
7. Pan JZ, Horrocks I, Schreiber G; OWL FA: A metamodeling extension of OWL DL. 2005; Proc International workshop on OWL: Experiences and Directions (OWL-ED2005):
8. Noy N. Representing classes as property values on the semantic web. 2005; 2005(March 2006): http://www.w3.org/TR/swbp-classes-as-values/.