Web Ontology Language: OWL

Grigoris Antoniou¹ and Frank van Harmelen²

¹ Department of Computer Science, University of Crete, ga@csd.uoc.gr
² Department of AI, Vrije Universiteit Amsterdam, Frank.van.Harmelen@cs.vu.nl

Summary. In order to extend the limited expressiveness of RDF Schema, a more expressive Web Ontology Language (OWL) has been defined by the World Wide Web Consortium (W3C). In this chapter we analyse the limitations of RDF Schema and derive requirements for a richer Web Ontology Language. We then describe the three-layered architecture of the OWL language, and we describe all of the language constructs of OWL in some detail. The chapter concludes with two extensive examples of OWL ontologies.

4.1 Motivation and Overview

The expressivity of RDF and RDF Schema that was described in [12] is deliberately very limited: RDF is (roughly) limited to binary ground predicates, and RDF Schema is (again roughly) limited to a subclass hierarchy and a property hierarchy, with domain and range definitions of these properties.

However, the Web Ontology Working Group of W3C³ identified a number of characteristic use-cases for Ontologies on the Web which would require much more expressiveness than RDF and RDF Schema.

A number of research groups in both America and Europe had already identified the need for a more powerful ontology modelling language. This lead to a joint initiative to define a richer language, called DAML+OIL⁴ (the name is the join of the names of the American proposal DAML-ONT⁵, and the European language OIL⁶).

DAML+OIL in turn was taken as the starting point for the W3C Web Ontology Working Group in defining OWL, the language that is aimed to be the standardised and broadly accepted ontology language of the Semantic Web.

In this chapter, we first describe the motivation for OWL in terms of its requirements, and the resulting non-trivial relation with RDF Schema. We then describe the various language elements of OWL in some detail.

³ http://www.w3.org/2001/sw/WebOnt/
⁵ http://www.daml.org/2000/10/daml-ont.html
⁶ http://www.ontoknowledge.org/oil/
Requirements for ontology languages

Ontology languages allow users to write explicit, formal conceptualizations of domains models. The main requirements are:

1. a well-defined syntax
2. a well-defined semantics
3. efficient reasoning support
4. sufficient expressive power
5. convenience of expression.

The importance of a well-defined syntax is clear, and known from the area of programming languages; it is a necessary condition for machine-processing of information. All the languages we have presented so far have a well-defined syntax. DAML+OIL and OWL build upon RDF and RDFS and have the same kind of syntax.

Of course it is questionable whether the XML-based RDF syntax is very user-friendly, there are alternatives better suitable for humans (for example, see the OIL syntax). However this drawback is not very significant, because ultimately users will be developing their ontologies using authoring tools, or more generally ontology development tools, instead of writing them directly in DAML+OIL or OWL.

Formal semantics describes precisely the meaning of knowledge. “Precisely” here means that the semantics does not refer to subjective intuitions, nor is it open to different interpretations by different persons (or machines). The importance of formal semantics is well-established in the domain of mathematical logic, among others.

One use of formal semantics is to allow humans to reason about the knowledge. For ontological knowledge we may reason about:

- **Class membership:** If $x$ is an instance of a class $C$, and $C$ is a subclass of $D$, then we can infer that $x$ is an instance of $D$.
- **Equivalence of classes:** If class $A$ is equivalent to class $B$, and class $B$ equivalent to class $C$, then $A$ is equivalent to $C$, too.
- **Consistency:** Suppose we have declared $x$ to be an instance of the class $A$. Further suppose that
  - $A$ is a subclass of $B \cap C$
  - $A$ is a subclass of $D$
  - $B$ and $D$ are disjoint

  Then we have an inconsistency because $A$ should be empty, but has the instance $x$. This is an indication of an error in the ontology.
- **Classification:** If we have declared that certain property-value pairs are sufficient condition for membership of a class $A$, then if an individual $x$ satisfies such conditions, we can conclude that $x$ must be an instance of $A$.

Semantics is a prerequisite for reasoning support: Derivations such as the above can be made mechanically, instead of being made by hand. Reasoning support is important because it allows one to