Reviewing the Design of DAML+OIL:  
an Ontology Language for the Semantic Web

Ian Horrocks a  Peter F. Patel-Schneider b  Frank van Harmelen c

a University of Manchester, Manchester, UK  b Bell Labs Research, Murray Hill, NJ, U.S.A.  c Vrije Universiteit, Amsterdam, Netherlands

Abstract

In the Semantic Web vision, the current “Syntactic Web”, where uninterpreted syntactic constructs are given meaning by private, inaccessible-to-computer agreements, is replaced by a web where both data and its semantic definition are accessible and manipulable by computer software. DAML+OIL is an ontology language specifically designed for this use in the Web; it exploits existing Web standards (XML and RDF), adding the familiar ontological primitives of object oriented and frame based systems, and the formal rigor of a very expressive description logic (DL). The logical basis of DAML+OIL means that reasoning services can be provided, both to support ontology design and to make Web data more accessible to automated processes. Since DAML+OIL has been designated as the foundation for OWL, the official W3C ontology-language standard for the Semantic Web, it is worth looking at considerations that justified the design of DAML+OIL.

The full version of this paper appeared in the proceedings of the Eighteenth National Conference on Artificial Intelligence, (AAAI’02), July 28 - August 1, 2002, Edmonton, Alberta, Canada.

Introduction

In the short span of its existence, the World Wide Web has resulted in a revolution in the way information is transferred between computer applications. It is no longer necessary for humans to set up channels for inter-application information transfer; this is handled by the TCP/IP and related protocols. It is also no longer necessary for humans to define the syntax and build parsers used for each kind of information transfer; this is handled by HTML, XML and related standards. However, it is still not possible for applications to interoperate with other applications without some pre-existing, human-created, and outside-of-the-web agreements as to the meaning of the information being transferred.

The next generation of the Web aims to alleviate this problem—making Web resources more readily accessible to automated processes by adding information that describes Web content in a machine-accessible and manipulable fashion. This coincides with the vision that Tim Berners-Lee calls the Semantic Web in his recent book “Weaving the Web”.

If such information (often called meta-data) is to make resources more accessible to automated agents, it is essential that its meaning can be understood by such agents. Ontologies will play a pivotal role here by providing a source of shared and precisely
defined terms that can be used in such meta-data. An ontology typically consists of a hierarchical description of important concepts in a domain, along with descriptions of the properties of each concept. The degree of formality employed in capturing these descriptions can be quite variable, ranging from natural language to logical formalisms, but increased formality and regularity clearly facilitates machine understanding.

The recognition of the key role that ontologies are likely to play in the future of the Web has led to the extension of Web markup languages in order to facilitate content description and the development of Web based ontologies, e.g., XML Schema, RDF (Resource Description Framework), and RDF Schema. RDF Schema (RDFS) in particular is recognisable as an ontology language: it talks about classes and properties, range and domain constraints, and subclass and subproperty relations.

RDFS is, however, a very limited language and more expressive power is clearly both necessary and desirable in order to describe data in sufficient detail. Moreover, such descriptions should be amenable to automated reasoning if they are to be used effectively by automated processes, e.g., to determine the semantic relationships between syntactically different terms. DAML+OIL is intended to extend the expressive power of RDFS, and to enable effective automated reasoning.

**Foundations in Description Logics** DAML+OIL is, in essence, equivalent to a very expressive Description Logic (DL). DAML+OIL classes can be names (URIs) or expressions, and a variety of constructors are provided for building class expressions. The expressive power of the language is determined by the class (and property) constructors provided, and by the kinds of axioms allowed.

**XML Schema datatypes** DAML+OIL supports the full range of XML Schema datatypes. This is facilitated by maintaining a clean separation between instances of “object” classes (defined using the ontology language) and instances of datatypes (defined using the XML Schema type system). In particular, the domain of interpretation of object classes is disjoint from the domain of interpretation of datatypes. The disjointness of object and datatype domains ensures that neither semantic integrity nor implementability are compromised by the introduction of the rather rich set of XML Schema datatypes.

**Extending RDF Schema** DAML+OIL is tightly integrated with RDFS: RDFS is used to express DAML+OIL’s machine readable specification,¹ and RDF provides the only serialisation for DAML+OIL. While the dependence on RDFS has some advantages in terms of the re-use of existing RDFS infrastructure and the portability of DAML+OIL ontologies, using RDFs to completely define the structure of DAML+OIL has proved quite difficult as, unlike XML, RDFS is not designed for the precise specification of syntactic structure.

**Inference in DAML+OIL** An important consideration in the design of DAML+OIL was that key inference problems in the language, in particular class consistency/subsumption, to which most other inference problems can be reduced, should be decidable, as this facilitates the provision of reasoning services. Moreover, the correspondence with DLs facilitates the use of DL algorithms that are known to be amenable to optimised implementation and to behave well in realistic applications in spite of their high worst case complexity.

¹[http://www.daml.org/2001/03/daml+oil.daml](http://www.daml.org/2001/03/daml+oil.daml)