OWL Web Ontology Language Summary

Introduction

OWL stands for Web Ontology Language. OWL has been developed by the Web Ontology Working Group as part of the W3C Semantic Web Activity for publication on 10 February 2004. It is one of the main components Semantic Web, which are about two things. It is about common formats for integration and combination of data drawn from diverse sources, where on the original Web mainly concentrated on the interchange of documents. It is also about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing. A simpler definition for OWL, that is a language that represents knowledge about a particular domain; it’s also being recognized as a specification of conceptualization.

In this summary, we will focus on OWL. OWL is used to publish and share sets of terms, supporting advanced Web search, software agents and knowledge management. It is also designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL can be used to explicitly represent the meaning of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called ontology. OWL facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics. OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full.

The OWL Language is described by a set of documents, each fulfilling a different purpose, and catering to a different audience. They are described and listed respectively as following in increasing degree of technical content.

- The **OWL Overview** gives a simple introduction to OWL by providing a language feature listing with very brief feature descriptions;
- The **OWL Guide** demonstrates the use of the OWL by providing an extended example. It also provides a glossary of the terminology used in these documents;
- The **OWL Reference** gives a systematic and compact (but still informally stated) description of all the modeling primitives of OWL;
• The **OWL Semantics and Abstract Syntax** document is the final and formally stated normative definition of the language;

• The **OWL Web Ontology Language Test Cases** document contains a large set of test cases for the language;

• The **OWL Use Cases and Requirements** document contains a set of use cases for a web ontology language and compiles a set of requirements for OWL.

**General Characteristics & Examples**

**Ontology structure:** OWL makes an open world assumption. That is, descriptions of resources are not confined to a single file or scope. While class C1 may be defined originally in ontology O1, it can be extended in other ontologies. The consequences of these additional propositions about C1 are monotonic. New information cannot retract previous information. New information can be contradictory, but facts and entailments can only be added, never deleted. In order to write an ontology that can be interpreted unambiguously and used by software agents we require a syntax and formal semantics for OWL. The OWL semantics are defined in OWL Web Ontology Language Semantics and Abstract Syntax.

**Elements:** Most of the elements of an OWL ontology concern classes, properties, instances of classes, and relationships between these instances.

- **Classes:** Classes provide an abstraction mechanism for grouping resources with similar characteristics. Like RDF classes, every OWL class is associated with a set of individuals, called the class extension. The individuals in the class extension are called the instances of the class. A class has an intensional meaning (the underlying concept) which is related but not equal to its class extension. Thus, two classes may have the same class extension, but still be different classes.

- **Properties:** Properties let us assert general facts about the members of classes and specific facts about individuals. **Object properties** link individuals to individuals. **Datatype properties** link individuals to data values.

- Property Characteristics: It specifies the property *characteristics*, which provides a powerful mechanism for enhanced reasoning about a property.
- Property Restriction: OWL further constrains the range of a property in specific contexts in a variety of ways.

**Ontology mapping:** In order for ontologies to have the maximum impact, they need to be widely shared. In order to minimize the intellectual effort involved in developing an ontology they need to be re-used. In the best of all possible worlds they need to be composed. For example, you might adopt a date ontology from one source and a physical location ontology from another and then extend the notion of location to include the time period during which it holds.

It is important to realize that much of the effort of developing an ontology is devoted to hooking together classes and properties in ways that maximize implications. This is the hardest part of ontology development. If you can find an existing ontology that has already undergone extensive use and refinement, it makes sense to adopt it.

**Ontology version:** Ontologies are like software, they will be maintained and thus will change over time. Within an element, it is possible to link to a previous version of the ontology being defined.

**Sublanguage Characteristics**

**OWL DL (Description Logic):** was designed to support the existing Description Logic business segment and to provide a language subset that has desirable computational properties for reasoning systems. It puts constraints on the mixing with RDF and requires disjointness of classes, properties, individuals and data values. The main reason for having the OWL DL sublanguage is that tool builders have developed powerful reasoning systems which support ontologies constrained by the restrictions required for OWL DL.

**OWL Full:** It’s the complete OWL language. It relaxes some of the constraints on OWL DL so as to make available features which may be of use to many database and knowledge representation systems, but which violate the constraints of Description Logic reasoners. It allows free mixing of OWL with RDF Schema and, like RDF Schema, does not enforce a strict separation of classes, properties, individuals and data values.

**OWL Lite:** It is a sublanguage of OWL DL that supports only a subset of the OWL language constructs. OWL Lite is particularly targeted at tool builders, who want to support OWL, but want to start with a relatively simple basic set of language features.
OWL Lite abides by the same semantic restrictions as OWL DL, allowing reasoning engines to guarantee certain desirable properties.

**Evolution**

OWL is developed as a vocabulary extension of RDF (the Resource Description Framework) and is also a revision of the DAML+OIL web ontology language incorporating lessons learned from the design and application of DAML+OIL.

The semantics have changed significantly. With respect to the three sublanguages, the DAML+OIL semantics are closest to the OWL DL semantics. There are major changes listed below:

1. The namespace was changed to http://www.w3.org/2002/07/owl.
2. Various updates to RDF and RDF Schema from the RDF Core Working Group were incorporated, including
3. Qualified restrictions were removed from the language.
4. Various properties and classes were renamed, as shown in the following table:
5. owl:SymmetricProperty was added.
6. Also added were owl:AnnotationProperty, owl:OntologyProperty and owl:DataRange.
7. An owl:DatatypeProperty may be an owl:InverseFunctionalProperty in OWL Full.
8. Synonyms for RDF and RDF Schema classes and properties were removed from the language, resulting in the removal of:
9. daml:disjointUnionOf was removed from the language, since it can be effected using owl:unionOf or rdfs:subClassOf and owl:disjointWith.
10. daml:equivalentTo was renamed to owl:sameAs, and is no longer a superproperty of owl:equivalentClass and owl:equivalentProperty.
11. owl:backwardCompatibleWith, owl: DeprecatedClass, owl:DeprecatedProperty, owl:incompatibleWith and owl:priorVersion were added to support versioning.
12. owl:AllDifferent and owl:distinctMembers were added to address the Unique Names Assumption.

**Views: Approval & Objection (concerns, objection)**

Adobe Systems, U.S. Department of Defense, Fujitsu Laboratories of America, HP Laboratories, IBM Research, Nokia, Sun Microsystems, Inc. etc. have adopted OWL
and given testimonials for it. These organizations are either the leading providers or top research laboratories. Some of them are W3C members. OWL is some of theirs business essence; some of them declare that such standardization of OWL will facilitate the acceptance of innovative software methods so on and so forth.

On the flip side, there are also several concerns and objections in real world practicing OWL. The first problem is that OWL does not allow for explicit declarations—assertions that a certain class, property, or an individual exists in an ontology. This aspect of the OWL standard was often misinterpreted, which caused design errors in OWL APIs; moreover, the lack of declarations makes devising an intuitive structural consistency check for OWL ontologies difficult. The second problem is that OWL Abstract Syntax and OWL RDF syntax rely on the separation between object and data property names for disambiguation. This prevents an unambiguous interpretation of certain syntactically well-formed OWL ontologies; furthermore, it makes implementing OWL parsers unnecessarily difficult. The third problem is that OWL Abstract Syntax cannot be translated into OWL RDF syntax without loss of information. Lastly, getting back to the core of the Semantic Web, how do we know we are now applying the good ontology?

**Conclusion/Current Status**

Web Ontology Working Group had announced by May 28, 2004 that the group officially came to the end. Also, they claimed they achieved what the group were chartered to do. They finally announced the success and concluded that RDF and OWL are Semantic Web standards that provide a framework for asset management, enterprise integration and the sharing and reuse of data on the Web.

**Reference**

[1] OWL Web Ontology Language [http://www.w3.org/TR/owl-features/](http://www.w3.org/TR/owl-features/)