O

OWL: Web Ontology Language

Adélia Gouveia
University of Madeira, Portugal

Jorge Cardoso
SAP Research CEC Dresden, Germany
University of Madeira, Portugal

INTRODUCTION

The World Wide Web (WWW) emerged in 1989, developed by Tim Berners-Lee who proposed to build a system for sharing information among physicists of the CERN (*Conseil Européen pour la Recherche Nucléaire*), the world's largest particle physics laboratory.

Currently, the WWW is primarily composed of documents written in HTML (hyper text markup language), a language that is useful for visual presentation (Cardoso & Sheth, 2005). HTML is a set of "markup" symbols contained in a Web page intended for display on a Web browser. Most of the information on the Web is designed only for human consumption. Humans can read Web pages and understand them, but their inherent meaning is not shown in a way that allows their interpretation by computers (Cardoso & Sheth, 2006).

Since the visual Web does not allow computers to understand the meaning of Web pages (Cardoso, 2007), the W3C (World Wide Web Consortium) started to work on a concept of the Semantic Web with the objective of developing approaches and solutions for data integration and interoperability purpose. The goal was to develop ways to allow computers to understand Web information.

The aim of this chapter is to present the Web ontology language (OWL) which can be used to develop Semantic Web applications that understand information and data on the Web. This language was proposed by the W3C and was designed for publishing, sharing data and automating data understood by computers using ontologies. To fully comprehend OWL we need first to study its origin and the basic blocks of the language. Therefore, we will start by briefly introducing XML (extensible markup language), RDF (resource description framework), and RDF Schema (RDFS). These concepts are important since OWL is written in XML and is an extension of RDF and RDFS.

BACKGROUND

Everyday, the Web becomes more attractive as an information sharing infrastructure. However, the vast quantity of data made available (for example, Google indexes more than 13 billion pages) makes it difficult to find and access the information required by the wide diversity of users. This limitation arises because most documents on the Web are written in HTML (HTML, 2007), a language that is useful for visual presentation but which is semantically limited. As a result, humans can read and understand HTML Web pages, but the contents of Web pages are not defined in a way that computers can understand them. If computers are not able to understand the content of Web pages it becomes impossible to develop sophisticated solutions to enable the interoperability and integration between systems and applications.

The aim of the Semantic Web is to make the information on the Web understandable and useful to computer applications and in addition to humans. *"The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation"* (Berners-Lee et al., 2001). The Semantic Web is a vision for the future of the Web, in which information is given explicit meaning, making it easier for machines to automatically process and integrate the information available on the Web.

One of the corner stones of the Semantic Web is the OWL. OWL provides a language that can be used by/on applications that need to understand the meaning of information instead of just parsing data for display purposes. Nowadays, several projects already rely on semantics to implement their applications. Example include semantic wikis (Campanini et al., 2004), social networks (Ding, et al., 2005), semantic blogs (Cayzer & Shabajee, 2003), and Semantic Web services (McIlraith et al., 2001),

THE SEMANTIC WEB STACK

The Semantic Web identifies a set of technologies and standards which form the basic building blocks of an infrastructure that supports the vision of the Web associated with meaning. Figure 1 illustrates the different parts of the Semantic Web architecture. It starts with the foundation of URI (universal resource identifier) and Unicode. URI is a formatted string that serves as a means of identifying abstract or physical resources. For example, http://dme.uma.pt/jcardoso/index.htm identifies the location from where a Web page can be retrieved and urn:isbn:3-540-24328-3 identifies a book using its ISBN. Unicode provides a unique number for every character, independent of the underlying platform, program, or language.

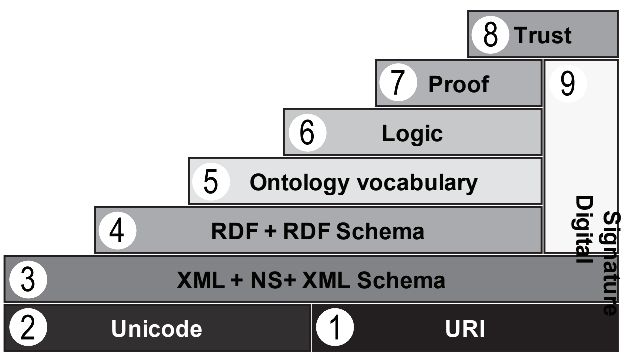
Directly above URI and Unicode we find the syntactic interoperability layer in the form of XML, which in turn underlies RDF and RDFS. Web ontology languages are built on top of RDF and RDFS. The last three layers are logic, proof, and trust, which have not been significantly explored. Some of the layers rely on the digital signature component to ensure security.

In the following sections we briefly describe the most relevant layers (XML, RDF, and RDFS). While the notions presented have been simplified, they give a reasonable conceptualization of the various components of the Semantic Web.

XML

The extensible markup language (XML) (Decker et al., 2000; XML, 2007) was originally pictured as a language for defining new document formats for the WWW. An important feature of this language is the separation of content from presentation, which makes it easier to select and/or reformat the data. SGML (standard generalized markup language) and XML are text-based formats that provide mechanisms for describing document structures using markup tags (words surrounded by '<' and '>'). Both HTML and XML representations use tags such as <h1> or <name>, and information between those tags, referred to as the content of the tag. However, there are significant differences between HTML and XML.

Figure 1. Semantic Web layered architecture (Berners-Lee et al., 2001)



XML is case sensitive while HTML is not. This means that in XML the start tags <Table> and <table> are different, while in HTML they are the same. Another difference is that HTML has predefined elements and attributes whose behavior is well specified, while XML does not. Instead, users can create their own XML vocabularies that are specific to their application or business' needs.

The following structure shows an example of an XML document identifying a 'Contact' resource. The document includes various metadata markup tags, such as <first_name>, <last_name>, and <e-mail>, which provides various details about a contact.

```
<Contact contact_id="1234">
    <first_name> Jorge </first_name>
    <last_name> Cardoso </last_name>
    <organization> University of Madeira </organization>
    <email> cardoso@uma.pt </email>
    <phone> +51 291 705 156 </phone>
</Contact>
```

While XML has gained much of the world's awareness, it is significant to identify that XML is simply a way of standardizing data formats. But from the point of view of semantic interoperability, XML has restrictions. One important characteristic is that there is no way to recognize the semantics of a particular domain because XML aims at a document structure and enforces no common interpretation of the data. Although XML is simply a data-format standard, it is part of a set of technologies that constitute the foundations of the Semantic Web.

RDF

Resource description framework (RDF) (RDF, 2002), was developed by the W3C to provide a common way to describe information so it could be read and understood by computer applications. RDF was designed using XML as the underlying syntax language. RDF provides a model for describing resources on the Web. A resource is an element (document, Web page, printer, user, etc.) on the Web that is uniquely identifiable by a URI. The RDF model is based upon the idea of making statements about resources in the form of a subject-predicate-object expression, a 'triple' in RDF terminology.

- Subject is the resource, that is, the thing that is being described;
- Predicates are aspects about a resource, and expresses the relationship between the subject and the object;
- Object is the value that is assigned to the predicate.

RDF has a very limited set of syntactic constructs, no other constructs except for triples is allowed. Every RDF