



Semantic Web: the devil is in the details

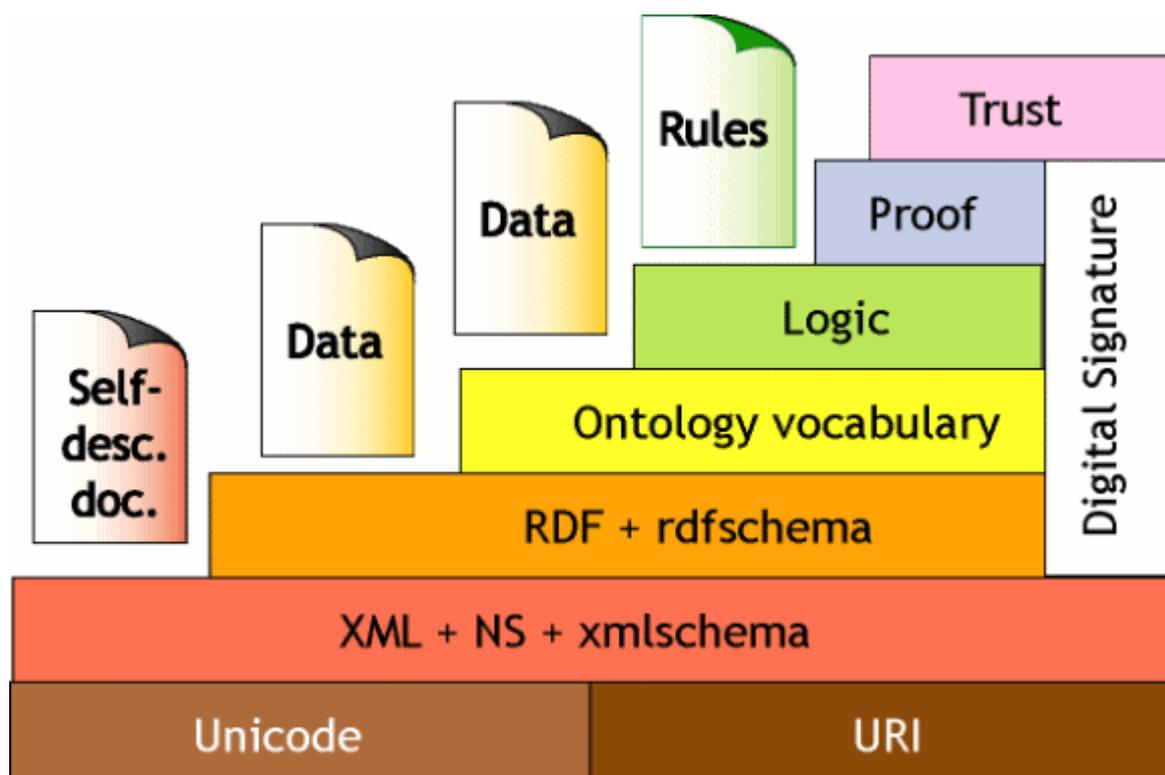
Cindy Meltzer

cmeltzer@hixnet.co.za

The vision of the semantic Web (SW) is to facilitate machine-readable data and to extend relational data to the Web through metadata that describe Web page data. The SW also makes metadata available to people and computers through a cycle of creation, discovery, integration and re-use. While the Web owes a lot of its success to the simplicity of design and application, the SW, although appearing simple at first, is considerably more complex. The final success of the SW will depend on the successful resolution of a number of issues. In this issue, some key concepts of the SW are briefly discussed. A gateway to sites dedicated to various aspects of the SW is also provided.

The SW is built as layers or levels of functionality in an architecture that bears similarity to the World-Wide Web architecture. In short, the SW assigns a uniform resource identifier (like a URL, but for data) to each relevant item and describes the relationship between the subjects and the properties in a resource description framework (RDF), using extensible markup language (XML) syntax. Tim Berners-Lee's articles on the SW design can be found at <http://www.w3.org/DesignIssues/Overview.html>. Figure 1 is Berners-Lee's 'layer cake' diagram of the SW architecture.

Figure 1 Diagram of the SW architecture (Berners-Lee)



Unicode and uniform resource identifier (URI) layers

Unicode and uniform resource identifier (URI) layers define international character sets and ensure unique addresses. The URI is the principal method for unique identification on the Web and is the means of identifying resources (or subjects) of the SW with their objects.

Extensible markup language (XML) layers: XML, namespace (NS) and XML schemas

Extensible markup language (XML) is the universal format for structured documents and data interchange on the Web. The W3C XML recommendation was first published on 10 February 1998 (<http://www.w3.org/TR/2000/REC-xml-20001006>) and, since then, XML has had widespread success with storing and exchanging data (<http://xml.coverpages.org/xmlApplications.html>). XML is publicly available, platform independent, simple, flexible and well-supported by major commercial vendors. These vendors have supported the development of XML as well as software that enables the use of XML. There are excellent resources on XML, including:

- Cover Pages (<http://coverpages.org>). This is a comprehensive Web-accessible reference collection that supports the standard general markup language (SGML) and XML family of (meta) markup language standards and their applications. The objectives of this public knowledgebase are to promote and enable the use of open, interoperable standards-based solutions that protect digital information and enhance the integrity of communication. The knowledgebase also aims to provide reference material on enabling technologies compatible with descriptive markup language standards and applications.
- XML.com (<http://www.xml.com/>). This is a comprehensive site that features a rich mixture of information and services for the XML community. The site is designed to serve both people who are already working with XML and hypertext markup language (HTML), as well as users who want to graduate to the power and complexity of XML.
- W3C XML homepage (<http://www.w3.org/XML/>). This site contains listed and linked working groups that find and download formal technical specifications and links to W3C recommendations, proposed recommendations, working drafts, conformance test suites and other documents.
- *An overview of XML*, an article by Margaret van Steenderen. The article can be found at http://general.rau.ac.za/infosci/raujournal/vol2.nr1.01_07_2000/default.asp?to=news.

To summarize, XML provides a *syntax* that allows a user-defined structure of a document in a standard, machine-readable form where the markup describes the structure of a document. XML documents can have any required elements. However, to define tag sets of all the elements and to attribute type names available to a type of document, users must either create a document type definition (DTD) that formally identifies the relationships between the various elements and attributes that form a document, or they must use a community standard DTD, for example, DocBook (<http://www.docbook.org/>) or an XML schema. An XML schema (<http://www.w3.org/XML/Schema>) describes a model for a whole group of documents. The model describes the possible arrangement of tags and text in a valid document. A schema might also be viewed as an agreement on a common vocabulary for a particular application that involves exchanging documents. It is therefore a collection (vocabulary) of type definitions and element declarations whose names belong to a particular XML namespace called a target namespace. Namespaces are the domain names associated with elements that produce a URI. This URI uniquely identifies the elements, which are central to both XML and RDF schemas. Target namespaces enable us to define groups of terms and to distinguish between definitions and declarations from different vocabularies.

XML schema languages such as XSD ([XML Schema Part 0: Primer](#)) can enable a simple to very sophisticated validation of the structure of an XML tree and can datatype validation of tag content. In combination with application profiles, this provides a certain amount of interoperability among a limited number of schemas, but it is not scalable to the entire World-Wide Web. The RDF layer provides the metadata interoperability required for data sharing.

RDF and RDF schemas

RDF is the metadata language that encodes metadata about resources on the Web. A resource is anything with an associated uniform resource identifier (URI). The RDF model consists of simple statements about resources. An URI is assigned to each relevant item (subject or property), and the relationship between subjects and properties is defined in a resource description framework (RDF). The RDF is encoded in XML.

RDF statements consists of three elements, namely:

- subjects or resources (anything that can be identified by a URI);
- predicates or properties (attributes to describe that resource); and
- objects (these values may be a literal or a second resource).

The vocabulary description language of RDF and RDF schemas is an extension of RDF. It provides mechanisms for describing groups of related resources and the relationships between these resources.

The starting site to visit is the W3C RDF home site, which contains reports, articles, developer tools, RDF validators and more. Go to <http://www.w3.org/RDF/>.

[Dave Beckett's Resource Description Framework \(RDF\) Resource Guide](#) is also an excellent site that contains RDF publications, documents, tools, tutorials and more.

Ontology layer

The ontology inference layer (OIL) is a proposal for a Web-based representation and inference layer for ontologies. On the SW, ontologies represent the semantics of documents and enable the semantics to be used by Web applications and intelligent agents. An ontology is a consensual, shared and formal description of the concepts that are important in a given domain. Typically, an ontology identifies classes of objects that are important in a domain and organizes these classes in a subclass hierarchy. Each class is characterized by properties that are shared by all elements in that class. Important relations between classes or between the elements of these classes are also part of an ontology. OIL synthesizes work from three different communities to achieve the aim of providing a general purpose markup language for the SW. OIL defines a number of classes and organizes them in a sub class hierarchy. The [Darpa Agent Markup Language](#) (DAML+OIL) is currently being developed into the [Web Ontology Language](#) (OWL) standard. WebOnt articulated the following design goals of a Web ontology, namely to:

- provide for ontology sharing;
- support evolution and revision within ontologies;
- support interoperability;
- detect inconsistencies in vocabularies;
- balance knowledge representation with scalability;
- provided ease of use;
- be serializable in XML; and

- pay due attention to the global character of the Web (<http://www.w3.org/TR/Webont-req/>).

The SW assumes that each ontology is consensual and congruent with the other shared ontologies to create a common domain of discourse that can be further interpreted by rules of inference and application logic. The current range of logic languages and information models have created a range of ontology forms, and the challenge is to create tools and ontologies capable of integration and interoperability.

On-to-knowledge project

The on-to-knowledge project aims to develop tools and methods for supporting knowledge management by relying on shareable and re-usable knowledge ontologies. The site provides information on OIL, papers, tool support and related initiatives. Go to <http://www.ontoknowledge.org/> for more information.

DAML ontology library

This library provides the URI of ontologies by keyword, URI and more. Go to <http://www.daml.org/ontologies/>.

Logic, proof and trust layers

Schemas and ontologies provide vocabularies and relationships between terms in vocabularies. They enable inference in the logical layer. The next step in the development of the SW is a logical language that makes inferences. The trust layer will include XML encryption and signatures. These are areas under development. See Berners-Lee's paper on design issues at <http://www.w3.org/DesignIssues/Logic.html>.

To process the knowledge available on the SW, an inference engine is necessary. Inference engines deduce new knowledge from already specified knowledge. There are two different approaches, namely general logic-based inference engines and specialized algorithms (problem-solving methods). The W3C XML query language is designed to provide a general purpose tool for querying XML documents. It will be defined in two parts, namely an [XML Query 1.0 and XPath 2.0 Data Model](#) and an [XQuery 1.0 Formal Semantics](#). Issues to be resolved include browser support when moving queries from clients to servers and the ability of the XML linking language to return queries made into RDF resources.

Digital signature layer

Digital signatures are a way of securely proving that someone wrote a specific document. It is an important layer in the 'Web of trust' without which information derived from the Web is not authenticated. [The XML signature syntax and processing W3C recommendation of 2002](#) specifies XML syntax and processing rules for creating and representing digital signatures. The recommendation points out that the XML signature is a method of associating a key with referenced data. It does not normatively specify how keys are associated with persons or institutions, nor does it specify the meaning of the data being referenced and signed. Consequently, while this specification is an important component of secure XML applications, it is not sufficient to address all application security and trust concerns, particularly with respect to using signed XML as a basis of human-to-human communication and agreement. Further security considerations must be taken into account. Go to <http://www.w3.org/TR/xmlsig-core> for more information.

Development

The SW is still in a developmental phase. Although rapid strides have been made in certain areas, other areas remain in a stage of possible change or have yet to be taken up. The SW community portal (<http://www.semanticWeb.org>) is the central resource for up-to-date information on all SW development and issues.

Problems that must be confronted include providing incentives for page creators to improve interoperability, as XML schemas with application profiles will provide a basis for internal organization, as well as data exchange between cooperating groups. Kendall Grant Clarke <http://www.xml.com/pub/a/2001/01/31/politics.html> has pointed out the political nature of the SW. According to Clarke, schemas, one of the foundational parts of the SW, are essentially political because they reflect the interests of the institutions that produce them. Other problems discussed include defining mappings between different ontologies for interoperability, the development of inference engines and security issues in the proof, trust and digital signatures layers.

The successful implementation of the SW will depend on a snowball effect that must begin with the development of tools, SW sites and standardization.

In 2002, the specification of the expression of simple Dublin Core Metadata in RDF and XML became a Dublin Core Metadata Initiative (DCMI) recommendation. Visit the Dublin Core Metadata Initiative site for a detailed discussion of this and other metadata issues at <http://dublincore.org/documents/2002/05/15/dcq-rdf-xml/>.

Interesting sites on the development of the SW include:

- The Maryland information and network dynamics lab SW agents project. This is Mindswap's first site on the SW and can be found at <http://www.mindswap.org>. This site harnesses many Web technologies (for example HTML and XML) and couples them with SW languages (RDF, RDFS, DAML+OIL, OWL) and tools that were built both here (see the download page) and elsewhere. Many of the pages contain links that allow you to either see the SW markup directly (the machine-readable markup) or to go to pages that describe how the pages are created and how the tools power them. This site has papers, demonstrations of Web service composers, DAML ontology search facilities, downloads of semantic markups, an ontology and RDF editor, an RDF instance creator and more.
- SWAD-Europe workplan. This site describes a number of reports, tools, projects and works in progress. Go to <http://www.w3.org/2001/sw/Europe/reports/intro.html> for more information.
- RDFLib.net is a [Python library](#) for working with RDF. RDFLib lets you generate, store and query RDF triples without requiring you to deal directly with RDF or XML syntax. It parses and serializes RDF and XML and stores RDF triples. Go to <http://rdflib.net/> for more information.
- The Open directory project uses RDF technologies. Go to <http://www.dmoz.org/>.
- Musicbrainz uses RDF and XML to facilitate the exchange of audio-video related metadata. Go to <http://www.musicbrainz.org/> for more information.
- RdfDB is a simple, scalable, open source database for RDF. Go to <http://www.guha.com/rdfdb/> for more information.
- SuishQL is a demonstrator, open source software, RDF query engine, written in Java. It can take standard query language-like query strings. Go to <http://swordfish.rdfWeb.org/rdfquery/> for more information.

Disclaimer

Articles published in SAJIM are the opinions of the authors and do not

necessarily reflect the opinion of the Editor, Board, Publisher, Webmaster or the Rand Afrikaans University. The user hereby waives any claim he/she/they may have or acquire against the publisher, its suppliers, licensees and sub licensees and indemnifies all said persons from any claims, lawsuits, proceedings, costs, special, incidental, consequential or indirect damages, including damages for loss of profits, loss of business or downtime arising out of or relating to the user's use of the Website.

ISSN 1560-683X

Published by [InterWord Communications](#) for the Centre for Research in Web-based Applications,
Rand Afrikaans University