World Academy of Science, Engineering and Technology
International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:3, No:10, 2009

International Science Index, Computer and Information Engineering Vol:3, No:10, 2009 waset.org/Publication/10028

The Semantic Web: a New Approach for Future World Wide Web

Sahar Nasrolahi[1], Mahdi Nikdast[2], Mehrdad Mahdavi Boroujerdi[3]

Abstract**—**The purpose of semantic web research is to transform the Web from a linked document repository into a distributed knowledge base and application platform, thus allowing the vast range of available information and services to be more efficiently exploited. As a first step in this transformation, languages such as OWL have been developed. Although fully realizing the Semantic Web still seems some way off, OWL has already been very successful and has rapidly become a defacto standard for ontology development in fields as diverse as geography, geology, astronomy, agriculture, defence and the life sciences. The aim of this paper is to classify key concepts of Semantic Web as well as introducing a new practical approach which uses these concepts to outperform Word Wide Web.

Keywords**—**Semantic Web, Ontology, OWL, Microformat, Word Wide Web.

I. INTRODUCTION

OVER the past decade, the Web has grown from what many perceived as an improved Gopher interface to became the new medium of communication. Today's World Wide Web is fundamentally a relatively simple artifact. Web content consists mainly of distributed hypertext, and is accessed via a combination of keyword based search and link navigation. This simplicity has been an important factor in Web's popularity and growth; naive users can easily use it and also are able to create their own content.

The current web can be characterized as the second Web generation: the first generation Web started with handwritten HTML pages; the second generation made the step to machine generated and often achieve HTML pages. These generations of the Web were meant for direct human processing (reading, browsing, form-filling). The third generation web, which one could call the "Semantic Web", aims at machine processable information. This coincides with the vision that Tim Berners-Lee describes in his recent book "Weaving the Web" [Berners-Lee, 1999]. The Semantic Web will enable intelligent services such as information brokers, search agents, information filters etc. Such intelligent services on the Knowledgeable Web should surpass the currently available versions of these services, which are limited in their functionality and most importantly only work as stand-alone services that do not interoperate.

The Semantic Web, with machine processable information contents, will only be possible when further levels of interoperability are established. Standards must be defined not only for the syntactic form of documents, but also for their semantic content. Such semantic interoperability is facilitated by recent W3C standardization efforts, notably XML/XML Schema and RDF/RDF Schema. These efforts are summarized in Fig.1.

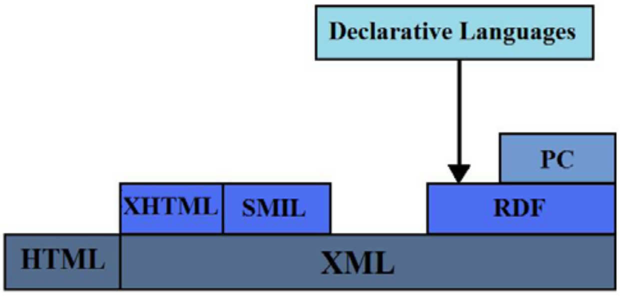


Fig. 1 Language Layers on the Web

The increase in both the range and quantity of Web content has, however, highlighted some serious shortcomings in the hypertext paradigm. In the first place, the required content becomes increasingly difficult to locate using the search and browse paradigm. Finding information about people with very common names (or with famous namesakes) can, for example, be a frustrating experience. More complex queries can be even more problematical. In the way of example a query for "animals that use sonar but are neither bats nor dolphins" may either return many irrelevant results related to bats and dolphins while the search engine failed to understand the negation or even may fail to return many relevant results because most relevant Web pages also mention bats and/or dolphins. More complex tasks may be extremely difficult or even impossible. Examples of such tasks include locating information in data repositories that are not directly accessible to search engines [2], or finding and using so-called web services [3].

If human users have difficulty accessing web content, the problem is even more severe for automated processes. This is because web content is primarily intended for presentation to and consumption by human users: HTML markup is mainly concerned with layout, size, color and other presentational issues. Moreover, web pages increasingly use images, often

[1, 2, 3] Department of Computer Engineering, Islamic Azad University, Najafabad Branch, Esfahan, Iran (e-mails: sahar.nasrolahi@gmail.com and mnikdast@ust.hk and mehrdad.mahdavi@live.com).

including active links, to present information. Human users are able to interpret the significance of such features, and thus understand the information being presented, but this may not be so easy for an automated process or "software agent". It is easy to imagine that similar difficulties might be experienced by users with cognitive or sensory impairments.

The Semantic Web aims to overcome some of the above mentioned problems by making web content more accessible to automated processes; the ultimate goal is to transform the existing web into "… a set of connected applications … forming a consistent logical web of data …" [4]. This is to be achieved by adding *Semantic Annotations* to Web content, for example annotations that describe the meaning of the content.

The rest of the paper is organized as follows: In section 2 we describe the background of the Semantic Web. The explanation of the important role of ontology in the architecture of the Semantic Web is done in section 3. Sections 4 briefly summarize the web ontology language OWL. In section 5 we introduce a new plug-in which has recently added to the Firefox internet browser with the aim of Semantic Web aspect. Finally we summarize our work in section 6.

II. BACKGROUND

As we mentioned above, the key idea behind the Semantic Web is to explicate the meaning of web content by adding semantic annotations. If we assume for the sake of simplicity that such annotations take the form of XML style tags, we could imagine a fragment of a web page being annotated as follows:

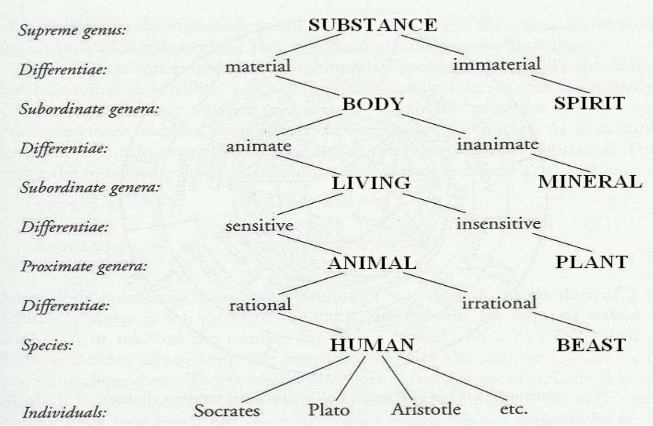


Fig. 2 Tree of Porphyry

<Wizard>Harry Potter</Wizard> has a pet called <SnowyOwl>Hedwig</SnowyOwl>

Taken in isolation, however, such annotations are of only limited value: the problem of understanding the terms used in the text has simply been transformed in to the problem of understanding the terms used in the labels. For example a query for information about raptors may not retrieve this text, even though owls are raptors. This is where *ontologies* come into play: they provide a mechanism for introducing a vocabulary and giving precise meanings to the terms in the vocabulary. A suitable ontology might, for instance, introduce the term SnowyOwl, and include the information that SnowyOwls are kinds Owl and that Owls are kinds of Raptor. Moreover, if this information is represented in a way that is accessible to our query engine, then it would be able to recognize that the above text is relevant to our query about raptors.

Ontology, in its original philosophical sense, is a fundamental branch of metaphysics focusing on the study of existence; its objective is to determine what entities and types of entities actually exist, and thus to study the structure of the world. The study of ontology can be traced back to the work of Plato and Aristotle, and from the very beginning included the development of hierarchical categorizations of different kinds of entity and the features that distinguish them: the well known "tree of Porphyry", for example, identifies animals and plants as sub-categories of living things distinguished by animals being sensitive, and plants being insensitive (see Fig.2).

III. ONTOLORY

Ontology in Semantic Web presents vocabularies and their relationships through the domain which they use. Fundamentally, ontology includes three main parts: concepts, relationships among concepts and finally their attributes. Fig.3 shows ontology's role in Semantic Web. The structure in this figure can be described as follow:

Vocabulary + Structure =Taxonomy
Taxonomy + Relationships, Constraints and Rules = Ontology
Ontology + Instances = Knowledgebase

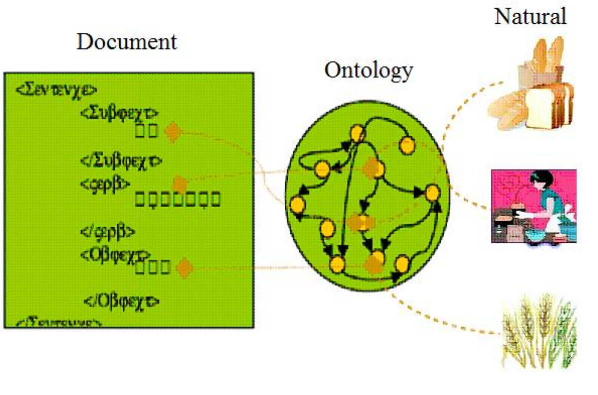
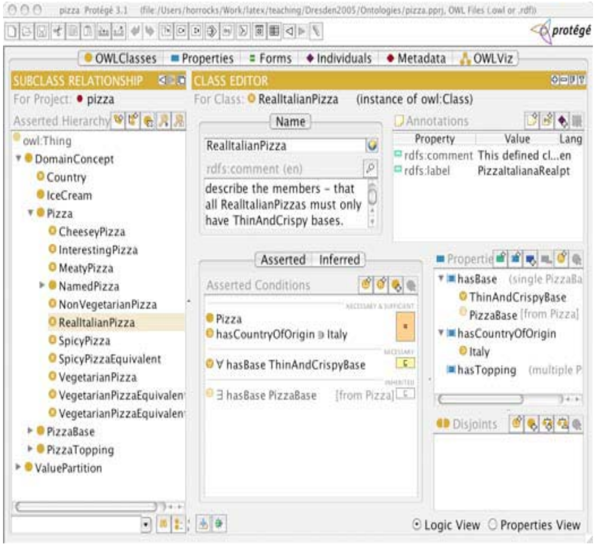


Fig. 3 The use of ontology in Semantic Web

In computer science, an ontology is usually taken to be a model of (some aspect of) the world; it introduces vocabulary describing various aspects of the domain being modeled, and provides an explicit specification of the intended meaning of the vocabulary. This specification often includes classification based information not unlike that in Porphyry's famous tree.

International Science Index, Computer and Information Engineering Vol:3, No:10, 2009 waset.org/Publication/10028

For example, Fig.4 shows a screenshot of a Pizza ontology as displayed by the Prot´eg´e ontology design tool [5]. The ontology introduces various pizza related vocabulary (some of which can be seen in the left hand panel), such as “NamedPizza” and “RealItalian- Pizza”, and arranges it hierarchically: RealItalianPizza is, for example, a sub-category of NamedPizza. The other panels display information about the currently selected vocabulary term, RealItalianPizza in this case, describing its meaning: a RealItalianPizza is a Pizza whose country of origin is Italy; moreover, a RealItalianPizza always has a ThinAnd- CrispyBase. Ontologies can be used to annotate and to organize data from the domain: if our data includes instances of RealItalianPizza, then we can return them in response to a query for instances of NamedPizza.



describe the members - that all RealItalianPizzas must only have ThinAndCrispy bases.

Fig. 4 Example Pizza Ontology

Ontologies (often also referred as Domain Model) can play a crucial role in enabling the processing and sharing of knowledge between programs on the Web. Ontologies are generally defined as a “representation of a shared conceptualization of a particular domain”. They provide a shared and common understanding of a domain that can be communicated across people and application systems. They have been developed in Artificial Intelligence to facilitate knowledge sharing and reuse.

An example of the use of ontologies on the Knowledgeable Web is in e-commerce sites, where ontologies are needed (a) to enable machine-based communication between buyer and seller, (b) to enable vertical integration of markets (e.g. www.verticalnet.com) and (c) to leverage reusable descriptions between different marketplaces.

A second example of the use of ontologies can be found in search engines. By using ontologies the search engines can escape from the current keyword-based approach and can find pages that contain syntactically different, but semantically similar words (e.g. www.hotbot.com).

Typically, an ontology contains a hierarchical description of important concepts in a domain (is-a hierarchy), and describes crucial properties of each concept through an attribute-value mechanism.

Additionally, further relations between concepts may be described through additional logical sentences. Finally, individuals in the domain of interest are assigned to one or more concepts in order to give them their proper type.

IV. THE WEB ONTOLOGY LANGUAGE OWL

Languages that are used for creating ontology can be classified into two main groups. Firstly, those which operate based on graph. Topic maps, UML, RDF Semantic Networks are some of the examples of this group. Secondly, languages which act according to the description logic. OWL belongs to the recent group.

The architecture of the Web depends on agreed standards such as HTTP that allow information to be shared and exchanged. A standard ontology language is, therefore, a prerequisite if ontologies are to be used in order to share and exchange meaning. Recognizing this fact, the World Wide Web Consortium (W3C) set up a standardization working group to develop such a language. The result of this activity was the OWL ontology language standard [6]. OWL exploited existing work on languages such as OIL [7] and DAML+OIL [8] and, like them, was based on a Description Logic (DL). In the following we will briefly introduce DLs and OWL. For more complete information the reader should consult The Description Logic Handbook [9], and the OWL specification [6].

A. Description Logic

Description logics (DLs) are a family of logic-based knowledge representation formalisms; they are descendants of Semantic Networks [10] and KL-ONE [11]. These formalisms all adopt an object-oriented model, similar to the one used by Plato and Aristotle, in which the domain is described in terms of individuals, concepts (usually called classes in ontology languages), and roles (usually called relationships or properties in ontology languages). Individuals, e.g., “Socrates” are the basic elements of the domain; concepts, e.g., “Human”, describe sets of individuals having similar characteristics; and roles, e.g., “hasPupil” describe relationships between pairs of individuals, such as “Socrates hasPupil Plato”.

As well as atomic concept names such as Human, DLs also allow for concept descriptions to be composed from atomic concepts and roles. Moreover, it is possible to assert that one concept (or concept description) is subsumed by (is a sub-concept of), or is exactly equivalent to, another. This allows for easy extension of the vocabulary by introducing new names as abbreviations for descriptions. For example, using standard DL notation, we might write:

$$HappyParent \equiv Parent \sqcap \forall\ hasChild\ .\ (Intelligent \sqcup Athletic)$$

This introduces the concept name HappyParent, and asserts that its instances are just those individuals that are instances of Parent, and all of whose children are instances of either intelligent or athletic.

Another distinguishing feature of DLs is that they are logics, and so have a formal semantics. DLs can, in fact, be seen as decidable subsets of first-order predicate logic, with individuals being equivalent to constants, concepts to unary predicates and roles to binary predicates. As well as giving a precise and unambiguous meaning to descriptions of the domain, this also allows for the development of reasoning algorithms that can be used to answer complex questions about the domain. An important aspect of DL research has been the design of such algorithms, and their implementation in (highly optimized) reasoning systems that can be used by applications to help them "understand" the knowledge captured in a DL based ontology.

A given DL is characterized by the set of constructors provided for building concept descriptions. These typically include at least intersection (⊓), union (⊔) and complement (¬), as well as restricted forms of existential (∃) and universal (∀) quantification, which in OWL are called, respectively, *someValuesFrom* and *allValuesFrom* restrictions. OWL is based on a very expressive DL called SHOIN that also provides cardinality restrictions (⩾,⩽) and enumerated classes (called *oneOf* in OWL) [12, 13]. Cardinality restrictions allow, e.g., for the description of a concept such as people who have at least two children, while enumerated classes allow for classes to be described by simply enumerating their instance, e.g.,:

$$EUcountries \equiv \{Austria,\ \ldots\ ,\ UK\ \}$$

SHOIN also provides for transitive roles, allowing us to state, e.g., that if x has an ancestor *y* and *y* has an ancestor *z*, then *z* is also an ancestor of *x*, and for inverse roles, allowing us to state, e.g., that if *z* is an ancestor of *x*, then *x* is also an descendent of *z*. The constructors provided by OWL, and the equivalent DL syntax, are summarized in TABLE I. In DLs it is usual to separate the set of statements that establish the vocabulary to be used in describing the domain (what we might think of as the schema) from the set of statements that describe some particular situation that instantiates the schema (what we might think of as data); the former is called the TBox (Terminology Box), and the latter the ABox (Assertion Box). OWL ontology is simply equivalent to a set of *SHOIN* TBox and ABox statements. This mixing of schema and data is quite unusual (in fact ontologies are usually thought of as consisting only of the schema part), but does not affect the meaning—from a logical perspective, *SHOIN* KBs and OWL ontologies are just sets of axioms.

TABLE I
OWL CONSTRUCTORS

Constructor	DL Syntax	Example
intersectionOf	C1 ⊓ . . . ⊓ Cn	Human ⊓ Male
unionOf	C1 ⊔ . . . ⊔ Cn	Doctor ⊔ Lawyer
complementOf	¬C	¬Male
oneOf	{x1 . . . xn}	{john, mary}
allValuesFrom	∀P.C	∀hasChild.Doctor
someValuesFrom	∃r.C	∃hasChild.Lawyer
hasValue	∃r.{x}	∃citizenOf.{USA}
minCardinality	(≥ n r)	(≥2 hasChild)
maxCardinality	(≤ n r)	(≤ 1 hasChild)
inverseOf	r−	hasChild−

The main difference between OWL and SHOIN is that OWL ontologies use an RDF based syntax intended to facilitate their use in the context of the Semantic Web. This syntax is rather verbose, and not well suited for presentation to human beings. E.g., the description of HappyParent given above would be written in OWL's RDF syntax as follows:

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="collection">
    <owl:Class rdf:about="#Parent"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasChild"/>
      <owl:allValuesFrom>
        <owl:unionOf rdf:parseType=" collection">
          <owl:Class rdf:about="#Intelligent"/>
          <owl:Class rdf:about="#Athletic"/>
        </owl:unionOf>
      </owl:allValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Fig.5 OWL's RDF Syntax

V. MICROFORMAT DETECTION EXTENSION

Today Mozilla Lab is realizing Operator, a microformat detection extension developed by Michael Kaply at IBM. Operator demonstrates the usefulness of semantic information on the Web, in real world scenarios. Operator leverages microformats and other semantic data that are already available on many web pages to provide new ways to interact with web services. Operator lets you combine pieces of information on Web sites with applications in ways that are useful. For instance, Flickr + Google Maps, Upcoming + Google Calendar, Yahoo! Local + your address book, and many more possibilities and permutations. All of these scenarios are possible due to Microformats, an emerging standard for injecting semantics into HTML.

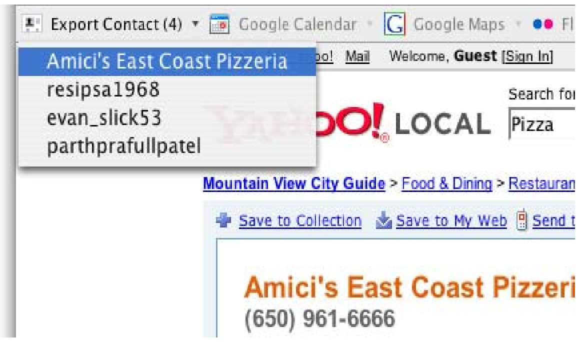


Fig. 6 Operator plug-in for Firefox

Operator requires information on the Web to be encoded using microformats, and since this method for semantically encoding information is relatively new, not all sites are using microformats yet. However, Operator works great with any blog that uses rel-tag, and the sites Yahoo! Local, Flickr, and Upcoming.org, all of which contain millions of pieces of information expressed using microformats. As more sites begin to semantically encode data with microformats, Operator will automatically work with them as well.

Operator isn't the first microformat detection extension for Firefox; previously microformat detection in Firefox was possible with the Tails Export extension by Robert de Bruin. Operator builds on Tails Export by having a user interface that is based around actions the user can take, instead of data types. Operator also includes support for the microformats geo and rel-tag, and is compatible with Firefox 2.

The combination of microformated content on the Web and the Operator extension for Firefox results in a kind of data cross pollination that we think is very exciting.

After using Operator for awhile, you will find yourself quickly transferring structured data to your favourite applications without typing a single letter, you will be hoping around the Web without navigating on hyperlinks, and you will be remixing services in ways that are really useful.

Here are some examples of things you can do with this release of Operator and with the Web as it exists today.

With Operator you can send the phone number of your favorite pizza place from Yahoo! Local to your address book, without having to type anything (see Fig.6.).

If you view an event at Upcoming.org you can easily add the event to your calendar to see if you are free, or map the location of the event to see where it will take place.

Fig. 7 Operator plug-in for Firefox

VI. CONCLUSION

In this paper we explain main conceptual issues of the Semantic Web as well as OWL ontology language. The ontology and its important role in the architecture of Semantic Web are purely discussed. The new generation of Web which has some Semantic aspects may lead to a more convenient platform for its users from all walks of life. The operator plug-in which has recently added to the Firefox browser is a small feature of such successes in Web. Analyzing and improving Description logic and languages which are used for creating ontology will be our future work.

REFERENCES

[1] I. Horrocks. Semantic Web: The Story So Far, 19th IEEE International Conference on Tools with Artificial Intelligence - Vol.1 (ICTAI 2007).

[2] R. Volz, S. Handschuh, S. Staab, L. Stojanovic, and N. Stojanovic. Unveiling the hidden bride: Deep Annotation for Mapping and Migrating Legacy Data to the Semantic Web. Journal of Web Semantics, 2004.

[3] S. McIlraith, T. C. Son, and H. Zeng. Semantic web services. IEEE Intelligent Systems, 16:46–53, 2001.

[4] T. Berners-Lee. Semantic web road map, Sept. 1998. Available at http://www.w3.org/DesignIssues/Semantic.html.

[5] H. Knublauch, R. Fergerson, N. Noy, and M. Musen. The Prot´eg´e OWL Plugin: An open development environment for semantic web applications. In S. A. McIlraith, D. Plexousakis, and F. van Harmelen, editors, Proc. of the 2004 International Semantic Web Conference (ISWC 2004), number 3298 in Lecture Notes in Computer Science, pages 229–243. Springer, 2004.

[6] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web Ontology Language semantics and abstract syntax. W3C Recommendation, 10 February 2004. Available at http://www.w3.org/TR/owlsemantics.

[7] D. Fensel, F. van Harmelen, I. Horrocks,D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. IEEE Intelligent Systems, 16(2):38–45, 2001.

[8] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In

Proc. Of the 18th Nat. Conf. on Artificial Intelligence (AAAI 2002), pages 792–797. AAAI Press, 2002.

[9] F. Baader, D. Calvanese, D. McGuinness, D.Nardi, and P. F. Patel-Schneider, editors. The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.

[10] W. A. Woods. What's in a link: Foundations for semantic networks. In R. J. Brachman and H. J. Levesque, editors, Readings in Knowledge Representation, pages 217–241. Morgan Kaufmann Publishers, San Francisco, California, 1985.

[11] R. J. Brachman and J. G. Schmolze. An overview of the Kl-One knowledge representation system. Cognitive Science, 9(2):171–216, Apr. 1985.

[12] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. J. of Web Semantics, 1(1):7–26, 2003.

[13] I. Horrocks and U. Sattler. A tableaux decision procedure for SHOIQ. In Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005), pages 448–453, 2005.

International Science Index, Computer and Information Engineering Vol:3, No:10, 2009 waset.org/Publication/10028