

OWL: Capturing Semantic Information using a Standardized Web Ontology Language

Aditya Kalyanpur, Jennifer Golbeck, Jay Banerjee, James Hendler
University of Maryland, College Park

The *Semantic Web* is an extension of the current World Wide Web. The hypertext pages that present information to humans remain, but a new layer of machine understandable data is added to allow computers to participate on the Web in new ways. Using standardized languages such as RDF and OWL, semantic web data can precisely describe the knowledge content underlying HTML pages, specify the implicit information contained in media like images and videos, or be a publicly accessible and usable representation of an otherwise inaccessible database.

An integral component of the Semantic Web is the notion of an ontology. Ontologies are also extensively used in natural language processing (NLP) systems. However, the lack of a standardized ontology language has made it difficult to share and reuse ontological information across interrelated systems. The Semantic Web provides such a standard – the Web Ontology Language (OWL) - which can be used to overcome the semantic interoperability problem, in addition to supporting a wide variety of intelligent web-based applications.

Ontologies in Natural Language Processing

Ontology has been an important concept in Philosophy, and later Library Sciences, for a long time before it became relevant to the Computer Science and in particular, the Artificial Intelligence (AI) community. In AI, an ontology is used to formally specify the concepts and relationships that characterize a certain body of knowledge (domain). The formal nature of ontologies makes them amenable to machine-readability and provides a well-defined semantics for the defined terms. This allows computer programs to manipulate, transform and draw inferences from information represented using the ontology.

Ontologies have been widely used in a variety of natural language applications including building a corpus of term definitions as a reference dictionary or thesauri (e.g. text classification systems), providing a systematic framework for complex language processing (e.g. word disambiguation based on context) and directly capturing rich linguistic knowledge (e.g. machine translation). A few examples of the kinds of the kinds of applications that contain ontological components include ¹:

- **Information Retrieval (IR):**
 - *cf.* CROSSMARC [Pazienza et al, 2003] is a European research project that emphasizes the utility of a generalized ontology architecture which decouples lexical and domain knowledge from conceptual knowledge, in order to facilitate multilingual information extraction across a diverse and dynamic web-based environment.
- **Question-Answering Systems:**
 - *cf.* AQUA [Vargas-Vera et al, 2004] is a question answering system that integrates several technologies such as ontologies (e.g. WordNet [Voorhees,

¹ Note: The examples are used to highlight the importance and role of ontologies in linguistic applications. A complete survey of ontology-based NLP tools is beyond the scope of this article

1993]), logic and NLP in order to improve user query resolution and precision of answers.

- **Machine Translation:**

- *cf.* OntoLearn [Gangemi et al, 2003] is a system used to translate multiword terms from English to Italian. The system automatically learns and extracts rich domain-specific ontologies from a corpus of text using complex semantic techniques, and uses the intermediate ontological representation to directly perform machine translation.

- **Language Understanding:**

- *cf.* IAMTC (Interlingual Annotation of Multilingual Text Corpora) [IAMTC] is a multi-site project focusing on the annotation of six sizable bilingual parallel corpora for interlingual content in order to support diverse NL applications. It uses the 110,000 node OMEGA ontology [Philpot et al, 2003] for part-of-speech tagging and natural language processing.

A Standardized Ontology Language

Until recently, the lack of a standardized ontology language has made it difficult to share and reuse ontologies across different applications within the same domain or across inter-related domains. Ontologies describing similar domain information varied significantly in syntax and semantics depending on the nature of the ontology language used. Hence, ontologies written in different languages needed to be modified and refined (usually manually) in order to borrow useful ontological data.

In order to address this interoperability problem and define a universal paradigm for web-based exchange of ontological information, the World Wide Web Consortium (W3C) created the Web Ontology Language (OWL) which became a W3C Recommendation in February of 2004. Using OWL as a common language knowledge experts and application developers can easily create, modify, link and import ontologies in a distributed environment. Accordingly, OWL is an important piece of the future vision of the web – the Semantic Web.

History of OWL

As noted earlier, ontologies existed outside the computer science community for a long time before they were aimed specifically at the Web starting in the mid 1990s. The first significant effort came in the form of the Simple HTML Ontology Extension [SHOE] language developed at the University of Maryland, which added tags necessary to embed arbitrary semantic data into WWW pages. Following on the heels of SHOE was the OIL [OIL] (Ontology Interchange Level) project, a University of Amsterdam led European Union (EU) initiative completed in 2000. The OIL language added description logic (DL) based formal semantics to frame-based models while being grounded in W3C standards such as XML and RDF (which itself had become a W3C recommendation in 1999).

Meanwhile, the DARPA Agent Markup Language Program (DAML) was launched in the US (Aug 2000) as a spin-off from Government investment in semantic web technology. It released DAML-ONT [DAML], an ontology specification language for expressing more sophisticated RDF class definitions than permitted by RDFS. The DAML project joined efforts with OIL

releasing DAML+OIL [DAML+OIL] in December 2000. DAML+OIL refined the original release, making the language's semantics more clear, and enabling the language to more successfully interoperate with DL-based tools.

Encouraged by the successful deployment and use of DAML+OIL and recognizing the need for a standardized ontology language that catered to the web's needs, the World Wide Web Consortium (W3C) formed the *Web Ontology Working Group* in Nov, 2002 aimed with the task of drafting the Web Ontology Language (OWL). The group was co-chaired by James Hendler (University of Maryland) and Guus Schreiber (University of Amsterdam) and consisted of 51 Members from 30 W3C Organizations, including prominent companies (HP, IBM and Sun etc) public sector agencies (DISA, NIST etc) and research labs (Stanford, UMD etc). OWL became a W3C recommendation in February 2004, and a set of documents describing the use and structure of the language can be found at <http://www.w3.org/2004/OWL>.

Characteristics of OWL

OWL is built on top of the Resource Description Framework (RDF) [RDF], which is itself built upon the XML syntax. RDF (including RDFS – the RDF schema language) and OWL provide the capability of creating Classes, Properties, and Instances. Classes (or concepts) are general categories that can be arranged in hierarchies. Each class defines a group of individuals that belong together because they share some properties. Instances (or individuals) are specific objects, and classes are used to define what type an object has. Properties (or relationships) are attributes of instances. Properties are defined generally and then used in instances to either specify data values or link to other instances.

To see how these three elements are used, consider the following example. We want to define a class of things called "People" and some properties of "People" like "name", "birthday", and "friend" (which will link a person to his or her friends). The RDF syntax to define the class and properties is as follows:

```
<Class ID="Person" />
<Property ID="name" />
<Property ID="birthday" />
<Property ID="friend" />
```

(The examples in this paper ignore one rather complex aspect of the semantic web, called namespaces, which is beyond the scope of this article – but basically they allow a term to be unambiguously designated as a web URI (for example <http://example.org/people.owl#Person>) instead of as an ungrounded natural language term.)

Once the class and properties are defined, they can be used to describe instances of the "Person" class. In this example, we will say that Joe Blog, born January 1, 1950, is friends with John Doe.

```
<Person ID="Joe">
  <name>Joe Blog</name>
  <birthday>January 1, 1950</birthday>
```

```

    <friend resource="#John" />
</Person>

<Person ID="John">
    <name>John Doe</name>
</Person>

```

There are a few interesting features to point out with this example. First, when describing Joe's friend John, the syntax changes to refer to a resource #John, rather than just putting John's name between the tags as we did with name and birthday. Using the resource construct allows us to say that Joe is friends with the person represented by the object named "John" that is defined elsewhere. This means that, instead of just listing data about Joe, we have now created a link between the two objects. This ability to link objects leads to the idea of an RDF graph, where objects are linked to their attributes and to each other.

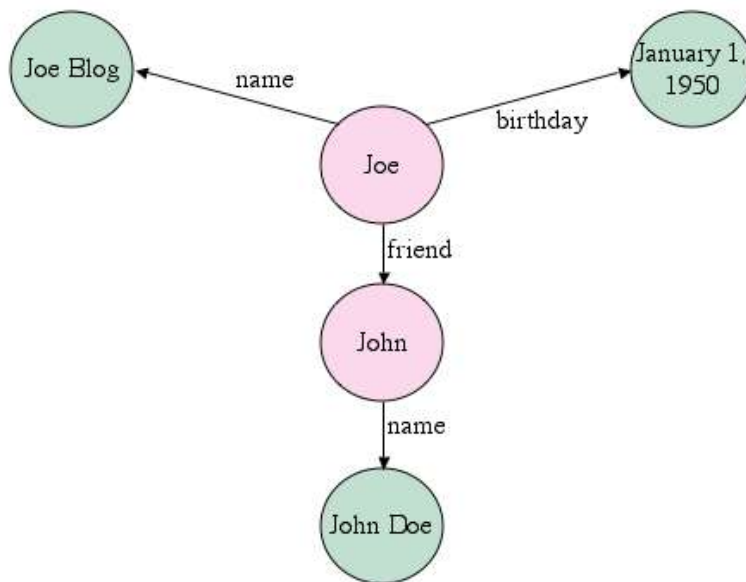


Figure 1: The RDF graph of the "Person" example.

The second point of interest is that we have assumed that the properties we defined are to be used with the "Person" class, but not with other classes we might define (we wouldn't want to put "friend" as a property of a "Chair" if we defined that class later on). At this point those are only assumptions. To enforce them, we can use the domain and range attributes to constrain the relationships. The domain let us limit the classes with which a property can be used and the range states which class an instance must come from to be used as a value. Example can thus be easily refined with domains and ranges.

```

<Class ID="Person" />

<Property ID="name">
    <domain resource="#Person" />
</Property>
<Property ID="birthday">

```

```
    <domain resource="#Person"/>
</Property>
<Property ID="friend">
    <domain resource="#Person"/>
    <range resource="#Person"/>
</Property>
```

OWL extends the functionality of RDF and RDFS considerably, while at its core, maintaining compatibility with the basic web architectural design i.e. it is open, non-proprietary, distributed across many systems, allows people to share data (ontologies), and scalable to Web-appropriate sizes. OWL provides three increasingly expressive sub-languages: OWL Lite, OWL DL, and OWL Full, each with a different intended audience based on scope and complexity of the application domain. For example, the goal of OWL Lite is to provide a language that is viewed by tool builders to be easy enough and useful enough to support, thereby acting as an entry ontology language for semantic web application developers, whereas OWL Full provides more freedom in domain modeling at the cost of a higher learning curve.

There are a several major capabilities that OWL adds to RDF and RDFS. The first is the ability to create local range restrictions. In RDF and RDFS, we could give one range for a property. However, in many cases the range for a property should vary based on what class is in the domain. For example, consider adding a property to our "Person" example called "eats". We would want to add a range restriction that the value for "eats" must come from a class of things called "Food". This is true for people in general. However, what if we create a subclass of "Person" called "Vegetarian". While vegetarians do eat food, they eat a more restrictive subset. However, using only RDF and RDFS, there is no way to say that people, in general, eat food while vegetarians do not eat meat. In OWL, we can leave the domain and range on the "eats" property unchanged, and state *in* the "Vegetarian" class that vegetarians are restricted to eating only vegetarian food (i.e. the value for the "eats" property must come from a class "VegetarianFood"). These types of restrictions on property ranges allow authors to create more expressive ontologies and for computers to make more logical assumptions. For example, if we know a person is a vegetarian, then we know anything they eat must be an instance of the "VegetarianFood" class.

OWL also introduces basic set functionality, such as unions, intersections, complements, and disjointness. Consider the case of defining subclasses of "Person" called "AlivePerson" and "DeadPerson". There is nothing stopping us from defining an object that is an instance of both the "AlivePerson" and "DeadPerson" class, because there is nothing defining these classes to be disjoint. In RDF and RDFS, there is nothing we can do about this, but in OWL we can state that the two classes are disjoint and ensure that an instance cannot be from both classes without the ontology becoming inconsistent.

One last addition worth mentioning is the introduction of cardinalities. OWL allows an author to put a restrictions on a property requiring it to be used on any instance a minimum number of times (minimum cardinality), a maximum number of times (maximum cardinality), or an exact number (cardinality). This means, for example, that we can require a *bilingual* person to speak at least two languages, define *twins* to be exactly two children, or allow a baseball team to have at most forty-four players on its roster.

There are other, more advanced features and intricacies of OWL beyond the scope of this article. Readers can visit the OWL project page at the World Wide Web Consortium (<http://w3.org/2004/OWL>) for an overview, extended examples and reference materials.

Applications of OWL

With the release of the OWL (and prior to that, RDF), language-related applications have begun to adopt semantic web technologies to enhance functionality and interoperability. An example is the knOWLer system [Ciorascu et al, 2003], an information management utility which demonstrates that ontological reasoning can scale to sizes of standard IR systems (100 million statements). The system supports sophisticated inferencing and query handling by using a description-logic based subset of OWL (with an ontology derived from WordNet). Another NL application that uses OWL is a machine translation system developed at Universiti Sains Malaysia [Lim et al, 2004], which performs word disambiguation using definition texts and structural information in an OWL ontology².

In addition, there have recently been some efforts in the semantic web community directly in support of language technologies. This includes a proposal to make small extensions to the RDF and OWL specs [Carroll et al, 2004] in order to efficiently build and maintain multilingual knowledge bases using the revised RFC 3066bis [Davis, 2003] specs that include productive use of language, country and script codes. Additional developmental efforts include SKOS-Core 1.0 [SKOS], an RDF vocabulary for representing in an online, machine-readable fashion. The goal is to provide an ideal entry point for the knowledge organization community to use standard semantic web technologies. Finally, the W3C's recently launched Semantic Web Best Practices and Deployment Working Group (SWBPD) has a WordNet task force [<http://www.w3.org/2001/sw/BestPractices/WNET/tf>] that among other things, aims to encourage dialog between Semantic Web developers and members of the lexical semantics community.

Aside from NLP systems, OWL has also been used in a number of diverse web-related applications. For example, OWL data is not restricted to simply augmenting HTML. It can be used to put a whole variety of data on the web that is not found in the text of pages. This includes, among other things, annotated photos ([Hollink et al, 2003], [Marques et al, 2003], [W3CPhoto]), social network representations [FoaF], and semantic web services [Sirin et al, 2003].

Another use of Semantic Web technologies has been in the design of Web portals. For example, figure 2 depicts a page at the MINDSWAP semantic web portal [<http://www.mindswap.org>]. It demonstrates how mainstream web technologies (HTML, XHTML, CSS) can be coupled with semantic web languages (RDF, OWL) to efficiently organize, filter and sort web content from diverse information sources. In this case, it provides all information it knows about the person *Jim Hendler* on a single page, by *semantically aggregating* annotated images containing the person, papers on which he is an author, news items that make a reference to him etc. At each

² The system uses the Protégé toolkit [<http://protege.stanford.edu/>] to develop and maintain the OWL ontology. Other notable examples of free and open-source ontology engineering tools are OilEd and SWOOP – more information about these and other tools can be found in a survey by Denny [2004]

point clear distinctions are made between his creations and creations made by someone else (i.e. what others have to say about him). This example serves to highlight the utility of the open, distributed and scalable nature of OWL in a rich and diverse information environment.

In summary, OWL provides a relatively rich semantics for defining on-line ontologies that are both powerful enough for contemporary natural language processing projects and consistent with modern web standards and architecture.



Figure 2: Semantic Web Page at MINDSWAP

References

- [Carroll et al, 2004] Jeremy J. Carroll and Addison Phillips "Multilingual RDF and OWL", Submitted to ISWC 2004
- [Ciorascu et al, 2003] Claudia Ciorascu Iulian Ciorascu, Kilian Stoffel. "knOWler - Ontological Support for Information Retrieval Systems" *In Proceedings of 26th Annual International ACM SIGIR Conference, Workshop on Semantic Web, Toronto, Canada, August 2003.*
- [DAML] DAML-ONT Initial Release Dec 2000. <http://www.daml.org/2000/10/daml-ont.html>
- [DAML+OIL] Deborah L. McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. "DAML+OIL: An Ontology Language for the Semantic Web ". *In IEEE Intelligent Systems, Vol. 17, No. 5, pages 72-80, September/October 2002..*
- [Davis, 2004] Phillips, A., Davis, M.: Tags for Identifying Languages. draft-phillips-langtags-02, 2004 (also known as RFC 3066bis)
- [Denny, 2004] A survey of Ontology Editors by Michael Denny (XML.com) http://www.xml.com/2004/07/14/examples/Ontology_Editor_Survey_2004_Table_-_Michael_Denny.pdf
- [FoaF] Friend of a Friend Project (FOAF): <http://www.foaf-project.org/>
- [Gangemi et al, 2003] A. Gangemi, R. Navigli, P. Velardi. "Corpus Driven Ontology Learning:

a Method and its Application to Automated Terminology Translation”, *IEEE Intelligent Systems*, January-February 2003, pp. 22-31

[**Hollink et al, 2003**] Laura Hollink, Guus. Schreiber, Jan Wielemaker and Bob. Wielinga. “Semantic Annotation of Image Collections”. In *S. Handschuh, M. Koivunen, R. Dieng and S. Staab (eds.): Knowledge Capture 2003 -- Proceedings-- Knowledge Markup and Semantic Annotation Workshop, October 2003*

[**IAMTC**] Multi-site NSF ITR project: Interlingual Annotation of Multilingual Text Corpora <http://aitc.aitcnet.org/nsf/iamtc/> (Members include: NMSU, CMU, USC, UMD, MITRE Corporation and Columbia University)

[**Lim et al, 2004**] Lian-Tze Lim and Tang Enya Kong. “Building an Ontology-based Multilingual Lexicon for Word Sense Disambiguation in Machine Translation”. In *Proceedings of the PAPILLON-2004 Workshop on Multilingual Lexical Databases Grenoble, August 30th-September 1st, 2004*

[**Marques et al, 2003**] Oge Marques & Nitish Barman. Semi-automatic semantic annotation of images using machine learning techniques In *Proceedings of the Second International Semantic Web Conference (ISWC), Oct 2003*

[**OWL**] OWL - Web Ontology Language Reference <http://www.w3.org/TR/owl-ref/>

[**Pazienza et al, 2003**] Maria Teresa Pazienza et al. "Ontology integration in a multilingual e-retail system" In *Proceedings of the HCI International, Heraklion, Crete, June 22-27, 2003*

[**Philpot et al, 2003**] Philpot, A., M. Fleischman, E.H. Hovy. 2003. “Semi-Automatic Construction of a General Purpose Ontology”. *Proceedings of the International Lisp Conference. New York, NY. Invited.*

[**RDF**] Revised RDF/XML Syntax Specification <http://www.w3.org/TR/rdf-syntax-grammar/>

[**RDFS**] RDF Vocabulary Description Language 1.0: RDF Schema <http://www.w3.org/TR/rdf-schema/>

[**SHOE**] Luke et al, SHOE 1.01 Specification, Apr 28, 2000

<http://www.cs.umd.edu/projects/plus/SHOE/spec.html>

[**Sirin et al, 2003**] Evren Sirin, Bijan Parsia, James Hendler. Semi-automatic Composition of web services using semantic descriptions. In *Proceedings of Web Services, Modeling, Architecture and Infrastructure workshop in ICIES 2003, Angers, France, April 2003*

[**SKOS**] Alistair Miles, Nikki Rogers, Dave Beckett. SKOS-Core 1.0 Guide: An RDF Schema for thesauri and related knowledge organisation systems.

<http://www.w3c.rl.ac.uk/SWAD/skos/1.0/guide/draft01.html>

[**Vargas-Vera et al, 2004**] Maria Vargas-Vera, Enrico Motta and John Domingue. “AQUA: An Ontology-Driven Question Answering System” In Maybury, Dr M. T., Eds. *Proceedings AAAI Spring Symposium, New Directions in Question Answering*, pages pp. 53-57, Stanford University, CA USA

[**Voorhees, 1993**] E. M. Voorhees. “Using wordnet to disambiguate word senses for text retrieval”. *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 171-180, PA, USA, 1993*

[**W3CPhoto**] W3C Photo Project: <http://www.w3photo.org/>