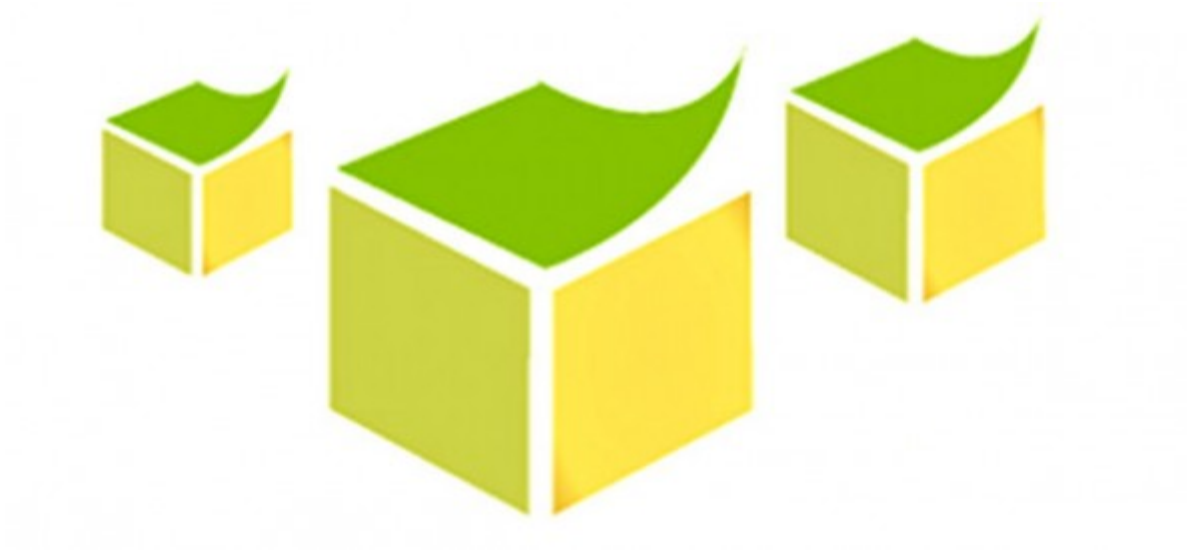
BUILDING THE SEMANTIC WEB



You might have come across the term "Semantic Web Applications" often, during talks about the future of Web apps. Check out what this is all about

---

There are two aspects to the possible future of the Web; one is that it is definitely going to be more social and more people-centric, but the other important characteristic is that it's going to be more semantic in nature. The Web technologies that we have available today focus more on how pages are displayed and what content is displayed, rather than on the content itself. The semantic Web is a facet of technology that will allow this content to become meaningful for machines, and will enable them to process this content and help us share, combine and analyse content effectively.

To make it clear, let me give you the example of Wikipedia. It has all the information in the world about everything related to rivers, but can I query it to find out about the longest river in India? I guess not, because I will have to search for it manually before retrieving that information, and there is no way to get this data programmatically from Wikipedia in its present form. Enter DBpedia, and you will be surprised to find that you can do much more than that. During the course of this article, we'll look at how this happens.

Why the Semantic Web?

But why go through all the hassle, you might ask? Well, natural language processing (NLP) is one field that could benefit from this. I'm not much of a fan of the Apple iPhone ecosystem, but we must agree that Siri is really an impressive feature. Having a machine that can easily interact with humans and answer random queries will be the next leap in the evolution of computing.

Another example where semantic information might come in handy is in search engines, which can then more effectively derive what the relationships between Web pages are, and understand their context, thus providing better search results that are relevant to the search query entered by the user. Google allows for retrieving such semantic information, as you can see in Figures 1 and 2.

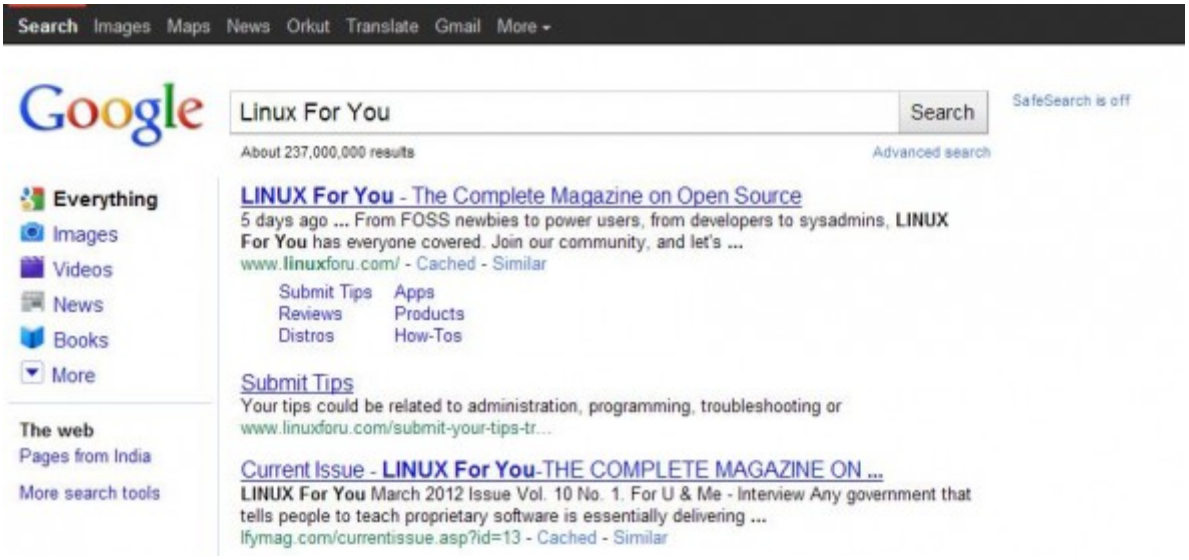


Figure 1: Google shows various parts of a website, retrieved with semantic information

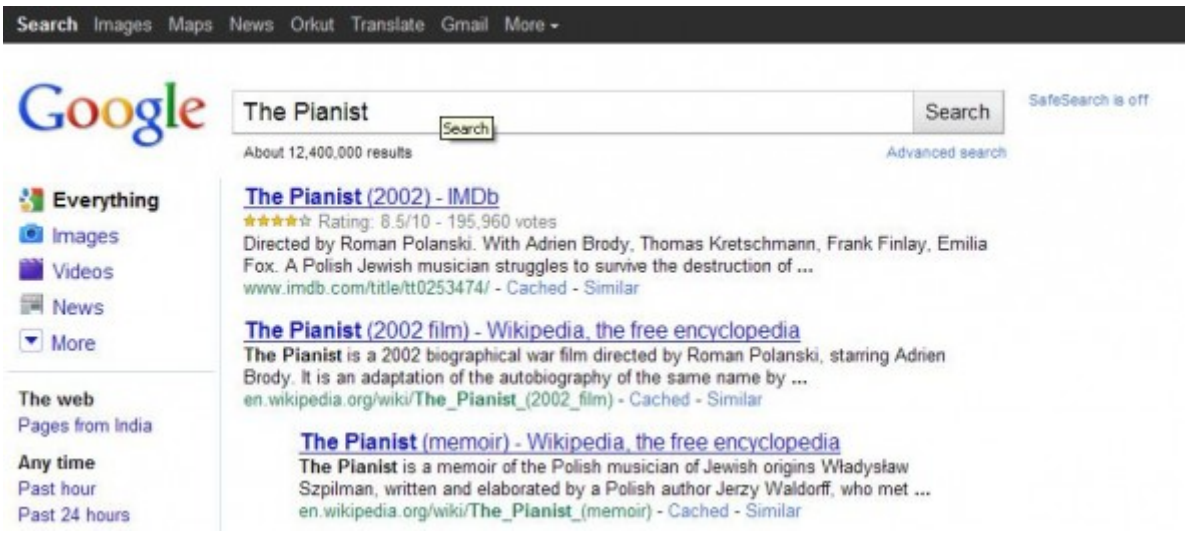


Figure 2: Movie ratings retrieved with semantic information

The Semantic Web stack

The Semantic Web stack is something that is more of a theoretical topic, and you'll find fat books on the topic. But when it comes to actually implementing it, it's a little more challenging than it sounds — and to put it into practice is a little overwhelming. Of course, we will look at the challenges that it can face, during the course of this article.

The various technologies that come under the umbrella of the "Semantic Web" have been under development, independently, for quite some time. Only after their recent standardisation have people realised that they are all part of the larger picture. Of these, the major technologies include:

- RDF (Resource Description Framework)
- RDFS (RDF Schema)
- OWL (Web Ontology Language)
- SPARQL (SPARQL Protocol and RDF Query Language)

These technologies collectively allow us to add semantic information to the vast number of existing Web pages. The aim recently has been to build upon the foundation of the Web as we know it, rather than to replace it altogether, because the former option is comparatively easier, when you consider the vast existing Web content that would be difficult to replace.

Resource Description Framework

RDF is a specification that deals with modelling or representation of data. Each set of information is represented in the form of subject-predicate-object syntax. There are a variety of formats in which this is serialised. Some of the popular ones are:

- RDF/XML
- RDFa
- N3
- Turtle

Again, these formats are only a subset of the many other notations available, which are under development. An example of how we would represent semantic data in the form of RDF/XML, which is the most common format, is as follows:

```
<?xml version="1.0"?><br>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:property="http://www.example.org/properties/"> <br>
     <rdf:Description
rdf:about="http://www.opensourceforu.com/semanticweb.html">
         <property:author>Ankit Mathur</property:author>
         <dc:language>en</dc:language>
         <dc:title>Semantic Web</dc:title>
</rdf:Description>
 </rdf:RDF>
```

Another possible notation, in RDF-Turtle, is as follows:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/properties/> .


< http://www.opensourceforu.com/semanticweb.html >
  Property:author "Ankit Mathur";
```

```
  dc:title "Semantic Web" ;
  dc:language "en";
  ] .
```

You can see that it’s easy to represent information about an article in many notations, though RDF/XML is a little more verbose if we compare them. But why did RDF come into the picture, when we could have done the same with XML and schemas?

Well, for one, XML is too flexible and schemas are too restrictive. When using XML schemas, we have to define the order and the possible values contained within the elements. Also, RDF can be used for annotation with documents, and is easily extensible, which is simply not possible with other technologies.

Resource Description Framework Schema

RDFS is basically used to describe properties and classes in RDF, so it is an extension that builds upon RDF itself. Here, I will quote the ever-popular example of cats and animals, where we can describe a Tiger, which is a type of animal, etc. This can be expressed in RDFS as follows:

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.example.com/animals#">
<rdfs:Class rdf:ID="animal" />
<rdfs:Class rdf:ID="tiger">
    <rdfs:subClassOf rdf:resource="#animal"/>
</rdfs:Class>
</rdf:RDF>
```

Ontologies: Web Ontology Language

Ontologies have often been talked about in computer science books. OWL, or Web Ontology Language, deals with the representation of these ontologies. But what exactly are ontologies? These are used to describe a domain of information and develop relationships between them — for example, classes, properties and their relationships. These ontologies are used to describe

and develop a semantic for each of the resources available on the Web, from which machine-processable information can be derived. This is just an enhanced form of what RDFS provides, and has more features and complexity. The OWL family consists of more than one sub-language and syntax. The sub-languages are:

- OWL Lite
- OWL DL
- OWL Full

To get a detailed overview of what OWL entails, you can pick up a book on ontologies and go through it first.

SPARQL

SPARQL is a query language used to retrieve information from the Semantic Web stack and its RDF representation. It is more verbose than SQL, and allows for more complex queries. There are various types of queries, depending on the usage:

- SELECT query
- DESCRIBE query
- ASK query
- CONSTRUCT query

In most cases, we have a SPARQL endpoint where SPARQL queries are processed, and results are returned in the form of XML, RDF or even HTML. Most queries in many applications can be constructed automatically. An oft-quoted example of a SPARQL query is as follows, which lists all landlocked countries with a population greater than 15 million:

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
    ?country a type:LandlockedCountries ;
             rdfs:label ?country_name ;
```

```
            prop:populationEstimate ?population .
    FILTER (?population > 15000000) .
}
```

Common vocabularies

There are a number of ontologies or vocabularies available online that allow you to describe common relationships among things, so that we do not need to describe them again and again. These are like a set of APIs that you can use. Some of the common vocabularies are FOAF (Friend of a Friend), SIOC (Semantically Interlinked Online Communities), SKOS (Simple Knowledge Organisation System), etc. As an example, let us look at FOAF — it allows you to describe relationships between people and their surroundings.

These are just some of the technologies that you will encounter while thinking about the Semantic Web. While developing real-world applications, you might come across RDF Triplestores, which could allow you to store semantic data and query it like a database. There are also Web frameworks like Jena, for server-side technologies like Java, which allow you to develop Semantic Web applications. These usually provide an abstraction API for all the above-mentioned technologies like RDF, OWL and SPARQL.

I hope this article has made you aware of what the upcoming technologies are in the field of Web development. This article was aimed more at making you aware of what's going on, rather than providing a tutorial for building a Semantic Web application.

Source : http://www.opensourceforu.com/2012/05/web-3-0-building-the-semantic-web/