What is the Semantic Web?

Currently the focus of a [W3C working group](http://www.w3.org/2001/sw), the Semantic Web vision was conceived by Tim Berners-Lee, the inventor of the World Wide Web. The World Wide Web changed the way we communicate, the way we do business, the way we seek information and entertainment — the very way most of us live our daily lives. Calling it the next step in Web evolution, Berners-Lee defines the Semantic Web as “a web of data that can be processed directly and indirectly by machines.”

In the Semantic Web data itself becomes part of the Web and is able to be processed independently of application, platform, or domain. This is in contrast to the World Wide Web as we know it today, which contains virtually boundless information in the form of documents. We can use computers to search for these documents, but they still have to be read and interpreted by humans before any useful information can be extrapolated. Computers can present you with information but can’t understand what the information is well enough to display the data that is most relevant in a given circumstance. The Semantic Web, on the other hand, is about having data as well as documents on the Web so that machines can process, transform, assemble, and even act on the data in useful ways.

Imagine this scenario. You’re a software consultant and have just received a new project. You’re to create a series of SOAP-based Web services for one of your biggest clients. First, you need to learn a bit about SOAP, so you search for the term using your favorite search engine. Unfortunately, the results you’re presented with are hardly helpful. There are listings for dish detergents, facial soaps, and even soap operas mixed into the results. Only after sifting through multiple listings and reading through the linked pages are you able to find information about the W3C’s SOAP specifications.

Because of the different semantic associations of the word “soap,” the results you receive are varied in relevance, and you still have to do a lot of work to find the information you’re looking for. However, in a Semantic Web-enabled environment, you could use a Semantic Web agent to search the Web for “SOAP” where SOAP is a type of technology specification used in Web services. This time, the results of your search will be relevant. Your Semantic Web agent can also search your corporate network for the SOAP specification and discover if your colleagues have completed similar projects or have posted SOAP-related research on the network. Based on the semantic information available for SOAP, your agent also presents you with a list of related technologies. Now you know that WSDL, XML, and URI are all technologies related to SOAP, and that you’ll need to do some research on them, too, before beginning your project. Armed with the information returned by your Semantic Web agent, you read the related technology specifications and send emails to the colleagues who have made SOAP-related materials available on the network to ask for their input before starting your new project.

Now, fast forward a few years. You’re still happily employed as a software consultant, and today you’re taking a working lunch with one of your biggest clients. Her company has an emergency project at its San Francisco branch for which they need you to consult for two weeks, and she asks you to get to San Francisco as soon as possible to begin work. You take out your hand held computer, activate its Semantic Web agent, and instruct it to book a non-stop flight to San Francisco that leaves before 10 AM the next day. You want an aisle seat if it’s available. Once your agent finds an acceptable flight with an available aisle seat, it books it using your American Express card and assigns the charges to your client’s account in your accounting application. It also warns you that you’ll be missing a dentist appointment back home during your trip and adds a note to your
calendar reminding you to reschedule. Next, you specify that you want a car
service to the client’s site, so your agent scans the availability of limos with “very
good” or higher service ratings and books an appointment to have you picked up
30 minutes after your flight lands. Your agent also books you at your favorite hotel
in San Francisco, automatically securing the lowest rate using your rewards card
number. Finally, the agent updates your calendar and your manager’s calendar
with your trip information and prints out your confirmation documents back at your
office.

With just a few clicks your Semantic Web agent found and booked your flight,
hotel, and car service, then updated your accounting system and calendars
automatically. It even compared your itinerary to your calendar and detected the
scheduling conflict with your dentist appointment. To do all this, the agent had to
find, interpret, combine, and act on information from multiple sources. This
example, of course, is a long-term vision for applying the Semantic Web. It’s one
that may or may not come to fruition, and only the future will tell. However, the
vision itself is important for understanding the potential of Semantic Web
technologies.

Considering the two examples above, the list of scenarios that could potentially
benefit from Semantic Web technologies as they continue to evolve is limited only
by the imagination. Think of the possibilities opened to everything from crime
investigation, scientific research, and literary analysis — to shopping, finding
long-lost friends, and vacation planning — when computers can find, present, and
act on data in a meaningful way.

The Semantic Web agent does not include artificial intelligence — rather, it relies on
structured sets of information and inference rules that allow it to “understand” the
relationship between different data resources. The computer doesn’t really
understand information the way a human can, but it has enough information to
make logical connections and decisions.

**Broadening Our Horizons**

The vision of the Semantic Web is a “web of data” that not only harnesses the
seemingly endless amount of data on the World Wide Web, but also connects that
information with data in relational databases and other non-interoperable
information repositories, for example, EDI systems. Considering that relational
databases house the majority of enterprise data today, the ability of Semantic
Web technologies to access and process it alongside other data from Web sites,
other databases, XML documents, and other systems increases the amount of
useful data available exponentially. In addition, relational databases already
include a great deal of semantic information. Databases are organized in tables
and columns based on the relationships between the data they house, and these
relationships reveal the meaning (the semantics) of the data.

Data integration applications offer the potential for connecting disparate sources,
but they require one-to-one mappings between elements in each different data
repository. The Semantic Web, however, allows a machine to connect to any other
machine and exchange and process data efficiently based on built-in, universally
available semantic information that describes each resource. In effect, the
Semantic Web will allow us to access all the information listed above as one huge
database.

**Defining Semantics and Relationships**

Implementing the Semantic Web requires adding semantic metadata, or data that
describes data, to information resources. This will allow machines to effectively
process the data based on the semantic information that describes it. When there
is enough semantic information associated with data, computers can make
inferences about the data, i.e., understand what a data resource is and how it
relates to other data.

XML (eXtensible Markup Language) has paved the road by adding some metadata
in the form of human-readable tags that describe data. In addition, XML
documents can include information about the author of a Web page, relevant
keywords for search engine optimization, and the software tools used to create the
XML file, for example.

Before XML, data was stored in flat file and database formats, where most data
was proprietary to an application. XML came along and made data interoperable
within a single domain, i.e., within the domain defined by a schema or a set of
related schemas. By itself, XML provides syntactic interoperability only when both
parties know and understand the element names used. If I label an element
What is the Semantic Web?

<price>12.00</price> and someone else labels it <cost>12.00<cost>, there’s no way for a machine to know that those are the same thing without the aid of a separate, highly customized application to map between the elements. Semantic Web technologies help address this problem by making tags understandable not just to humans – but to machines as well.

The first step required for machines to understand data is to get that data into a uniform format, where, for instance, a field labeled “street” always has the same format and contains the same type of information, and so on. This type of functionality can be found today on Web sites that use forms that allow users to enter information and run a query, such as airline Web sites that allow visitors to search for and book flights based on a variety of criteria. However, considering the amount and variety of data available from different sources today, this method of data typing does not scale beyond very specific applications.

The next step towards the Semantic Web requires that data from multiple domains is classified based on its properties and its relationship with other data. This is where Semantic Web technologies such as RDF, RDFS, and OWL come in.

Resource Description Framework (RDF)

An official W3C recommendation, RDF is an XML-based standard for describing resources that exist on the Web, intranets, and extranets. RDF builds on existing XML and URI (Uniform Resource Identifier) technologies, using a URI to identify every resource, and using URIs to make statements about resources. RDF statements describe a resource (identified by a URI), the resource’s properties, and the values of those properties. RDF statements are often referred to as “triples” that consist of a subject, predicate, and object, which correspond to a resource (subject) a property (predicate), and a property value (object). Below is an example of an RDF statement in plain English:

```
[resource] [property] [value]
The secret agent is Niki Devgood
[subject] [predicate] [object]
```

RDF triples can be written with XML tags, and they are often conceptualized graphically as shown below:

![RDF Graph](http://www.altova.com/semantic_web.html)

After creating this triple, we can go on to create other triples to associate the agent with an email address, image, etc.

Once triples are defined graphically, they can be coded in either RDF/XML or n-Triples formats to be accessed programmatically.

By creating triples with subjects, predicates, and objects, RDF allows machines to make logical assertions based on the associations between subjects and objects.
And since RDF uses URIs to identify resources, each resource is tied to a unique definition available on the Web. However, while RDF provides a model and syntax (the rules that specify the elements of a sentence) for describing resources, it does not specify the semantics (the meaning) of the resources. To truly define semantics, we need RDFS and OWL.

RDF Schema (RDFS)

RDFS is used to create vocabularies that describe groups of related RDF resources and the relationships between those resources. An RDFS vocabulary defines the allowable properties that can be assigned to RDF resources within a given domain. RDFS also allows you to create classes of resources that share common properties.

Using the same triples paradigm defined by RDF, RDFS triples consist of classes, class properties, and values that define the classes and relationships between the resources within a particular domain.

In an RDFS vocabulary, resources are defined as instances of classes. A class is a resource too, and any class can be a subclass of another. This hierarchical semantic information is what allows machines to determine the meanings of resources based on their properties and classes.

Below is a graphical example of an RDFS that shows a resource and its associated properties, values, and classes.

Overall, RDFS is a simple vocabulary language for expressing the relationships between resources. Building upon RDFS is OWL, which is a much richer, more expressive vocabulary for defining Semantic Web ontologies.

Web Ontology Language (OWL)

OWL is a third W3C specification for creating Semantic Web applications. Building upon RDF and RDFS, OWL defines the types of relationships that can be expressed in RDF using an XML vocabulary to indicate the hierarchies and relationships between different resources. In fact, this is the very definition of "ontology" in the context of the Semantic Web: a schema that formally defines the hierarchies and relationships between different resources. Semantic Web ontologies consist of a taxonomy and a set of inference rules from which machines can make logical conclusions.

A taxonomy in this context is system of classification, such as the scientific
What is the Semantic Web?

http://www.altova.com/semantic_web.html

kingdom/phylum/class/order/etc. system for classifying plants and animals that groups resources into classes and sub-classes based on their relationships and shared properties.

Since taxonomies (systems of classification) express the hierarchical relationships that exist between resources, we can use OWL to assign properties to classes of resources and allow their subclasses to inherit the same properties. OWL also utilizes the XML Schema datatypes and supports class axioms such as subClassOf, disjointWith, etc., and class descriptions such as unionOf, intersectionOf, etc. Many other advanced concepts are included in OWL, making it the richest standard ontology description language available today.

A graphical example of an OWL ontology is below.

All the detailed relationship information defined in an OWL ontology allows applications to make logical deductions. For instance, given the ontology above, a Semantic Web agent could infer that since "Goose" is a type of "DarkMeatFowl," and "DarkMeatFowl" is a subset of the class "Fowl," which is a subset of the class "EdibleThing," then "Goose" is an "EdibleThing."

It’s important to note that OWL has three sub languages, each with increasing complexity: OWL Lite, OWL DL, and OWL Full. OWL DL includes OWL Lite, and OWL Full includes OWL DL and OWL Lite. Developers choose which OWL dialect to use based on the level of complexity and level of detail required by their semantic model.

When RDF resource descriptions are associated with an ontology defined somewhere on the Web, intranet, or extranet, it’s possible for machines to retrieve the semantic information associated with each resource. It’s in this way that URIs, XML, RDF, RDFS, and OWL combine to make the Semantic Web a reality, making scenarios such as the software consultant’s SOAP research and business trip planning described earlier feasible.

Other Semantic Web Technologies

In addition to the technologies mentioned here, provisions are also in place for Semantic Agent "proofs," which allow humans to retrace the steps a Semantic Web agent took to arrive at a particular conclusion, as well as for security and trust mechanisms provided through digital signatures.

Also, as mentioned earlier, Semantic Web agents, which are computer programs capable of interpreting RDF and OWL semantic information, are also required for harnessing the power of the Semantic Web.

For more information about the Semantic Web and its associated technologies, visit the links at the end of this page.

Semantic Web Present and Future

It’s important to note that implementation of RDF, OWL, and the Semantic Web as a whole will be a gradual process. Questions about what the Semantic Web is and how it can benefit businesses and individuals are similar to initial confusion about
What is the Semantic Web?

why we needed HTTP and the Web before “WWW” was a staple of our daily vocabulary. But considering how those technologies have proliferated, it’s likely that the Semantic Web vision is one that will be realized, even if it’s on a small scale initially.

It’s also important to note that, similar to current Web services implementations, the Semantic Web may initially be restricted to intranet and extranet applications until questions about information security can be sufficiently addressed.

The true impact of the Semantic Web will not be known for quite some time, but its potential is staggering. Some Semantic Web proponents have asserted that it will lead to the evolution of human knowledge itself by allowing people - for the first time - to quickly filter and synergize the massive amounts of data that exist in the world in a relevant, productive way.

Visual Semantic Web Development

Given that RDF, RDFS, and OWL Semantic Web documents are often represented graphically, it makes sense to develop the corresponding RDF/XML or N-Triples code in a highly-visual manner. Following in its tradition of supplying developers with easy-to-use, visual development tools, Altova created SemanticWorks™ 2006 to help our customers learn and work with these new Semantic Web technologies in an intuitive way.

A visual RDF and OWL editor, SemanticWorks includes the following powerful functionality:

* Support for visual creation and editing of RDF, RDF Schema (RDFS), OWL Lite, OWL DL, and OWL Full documents
* Intelligent entry helpers that offer context-sensitive editing choices
* Syntax checking for RDF, RDFS, and OWL documents
* Semantics checking for OWL Lite and OWL DL ontologies
* Auto-generation and editing of RDF/XML or N-triples formats based on visual RDF/OWL design

Altova SemanticWorks™ 2006 allows you to graphically create and edit RDF instance documents, RDFS vocabularies, and OWL ontologies with full syntax checking and ontology semantics checking. Context-sensitive entry helpers present you with a list of permitted choices based on the RDF or OWL dialect you’re using, so you can create valid documents quickly and easily.

You can switch from the graphical RDF/OWL view to the text view to see how your document is being built in RDF/XML or N-triples format, and you can export your file from RDF/XML to N-triples or vice versa at any time. And, because the RDF/XML or N-triples code is auto-generated based on your design, you can learn and experiment with the concepts of the Semantic Web without having to write complicated code.
Learn more about working with the Semantic Web using Altova SemanticWorks™ 2006 [here](http://www.altova.com/semantic_web.html). Or, get started right away — download a free 30-day trial of SemanticWorks now.

Resources for Further Information

- Altova SemanticWorks™ 2006: Example RDF, RDFS, and OWL files are shipped with the product, and the help file included with SemanticWorks contains an informative tutorial to get you started learning RDF and OWL quickly and easily. SemanticWorks™ 2006 is available as a free 30-day trial [download](http://www.altova.com/semantic_web.html).
- SemanticWeb.org
- How the Semantic Web Works
- Semantic Web in Breadth
- What is RDF?
- RDF FAQ (W3C)
- W3C RDF Primer
- W3 Schools RDF Tutorial
- W3 Schools OWL Tutorial
- Description of OWL