Semantic Web: the Future of the Internet

The Internet has revolutionised both the IT sector and the lives of millions of people all across the world. However, its rapid development has also brought some of its weaknesses to the fore. One of these was made brutally clear to me by my teenage sister, who recently raged at the uselessness of web browsers. Though usually an avid web user, this time she flatly stated that the Internet was stupid and useless. And you know what brought this frenzy on? She could not find pictures illustrating a physics experiment for measuring the speed of light (she needed them for a physics project at school). My sister complained that you cannot find anything on the Internet (like I could do something about it). And why cannot you just type in that you need such and such pictures and just get them on the screen? What could I say...

Growing problems with finding information of the web are well known to us all, as they affect the majority of Internet users. There are many reasons for this state of affairs, but they all boil down to one fundamental premise: computers do not understand the information flowing through the Internet, only humans do. How then can information be searched for effectively if, from the point of view of search engines, it does not actually exist? After all, a web search typically involves searching websites and documents for occurrences of specified keywords, treated as character strings with no special meaning, even though searching for information is a process where meaning (semantics) should be of paramount importance. Search engines make no allowance for the context, although this can radically change the meaning of homonyms, which are words with the same spelling for several meanings, such as lead or ball. Traditional search engines also fail to account for synonyms, i.e. different words with similar or identical meanings. For example, searching for documents on friars should yield the same results as searching for documents on monks. All these problems are to be resolved by the Semantic Web, which will make information meaningful to machines by providing a formalised semantic description language.

The author works at ASTEC, a company which is part of the ASG (Adaptive Services Grid) project development consortium.
Contact with the author: autorzy@software.com.pl

Figure 1. Diagram showing a part of the journey ontology

The Semantic Web – the Shape of Tomorrow

The Semantic Web takes the Internet to the next level and will not only make it possible to run semantic searches, but can potentially revolutionise the entire browsing experience. Searching and navigating websites will come down to formulating requests in our natural language. The technology to make this possible is called Semantic Web Services (SWS).

To fully appreciate the power of the SWS approach, consider the following example. Imagine you want to travel from A (where you live) to B, book a room in the cheapest hotel in B and rent a car there. With current web technologies, this would be a pretty complicated task. A typical scenario would start with finding out how you can get from A to B, deciding on a carrier and finding its website. Once you have bought or reserved a ticket, you can go on to finding a hotel. This would require you to find the websites of all the hotels in B, compare their offers and decide on the cheapest one. The same procedure then applies to renting a car. All this is rather tedious, especially as you do not know which actual websites to visit and have to find them first. Besides, every online reservation requires you to fill out an almost identical form with such information as your name, address, e-mail, etc., so the whole process takes up a great deal of precious time.

If we had the Semantic Web at our disposal, the scenario would be radically different. All you would need to do is open up a semantic search engine and enter a query like: Find a means of transport to town B, buy a ticket for it, reserve a room in the cheapest hotel in the centre of B and rent a car. And that is it. Powerful mechanisms
would then swing into action to process your query and make sure your requirements are fulfilled as completely as possible. This is precisely the sort of functionality that the Semantic Web Services, currently under development, are supposed to offer. I think you will agree it is an appealing idea, even though it borders slightly on science fiction. How, then, can SWS be implemented in practice?

To answer this question, we first need to define the problems to overcome. The first is that the majority of information on the web is useful primarily to humans, and only in a limited degree to applications. The other problem is that of communication, both between man and machine and between applications. The ultimate aim is to enable humans to communicate their requests in natural language, while also allowing various applications using SWS to exchange information without human participation. This would make it possible to automatically find and call services, as well as pass results from one application to another, which in turn would facilitate complex queries requiring the cooperation of several SWS sites. However, communication requires a common vocabulary, which would enable all parties to agree on the same interpretation of a message. If I say the word *club*, then all the participants of a dialogue should correctly interpret the meaning of this term. In theory, everyone knows what the word means, but in reality its informational content depends on the context. The word *club* can, for example, mean a recreational building, a blunt tool used as a weapon or a team of football players – the problem was mentioned in this article. Fear not, for help is at hand in the unexpected form of ontology.

**Ontology: a Visitor from the Realm of Philosophy**

First, let us establish what ontology actually is and see if it is as distant from programming and IT as it may at first appear.

The term itself was borrowed from philosophy, where it signifies studies on the most fundamental of issues: existence itself and the character and structure of reality.

In IT, the word *ontology* means a representative vocabulary of terms for a selected area of knowledge (recreation, dentistry, computer science, etc.). An ontology in the IT sense provides a set of names for elements of a select- ed segment of reality (classes, associations and relations) along with descriptions of their meaning and constraints of their interpretation. Ontologies couple knowledge bases with reasoning rules by introducing a twofold structure into the vocabulary: relations and term hierarchies. Ontologies are used by humans, applications and knowledge bases to share a certain area of information, and shared knowledge is a necessary prerequisite of communication. Without a common knowledge base, mutual understanding is not possible.

An ontology could also be seen as a meta-model (formulated for example in UML notation) of a segment of reality, which includes all the objects and object interrelations required to describe this segment, along with property values (attributes) supplied by the user. Figure 1 shows a part of the *journey* ontology (in UML notation), which would be necessary for planning a journey. As you can see, the ontology defines the subject *me* representing the user, a class called *ticket* and relations between the *ticket* element and the subject, such as *book* or *buy*. The ontology also introduces a hierarchy by creating two new classes, *railway ticket* and *airline ticket*, from the base class *ticket*. What is not shown in Figure 1 are constraints formulated in a selected formal language of logic (such as OCL or F-logic) and describing the classes and relations. One such constraint could be a clause specifying that the *buy* relation is permissible only if the subject *me* has enough money to pay for the ticket. An ontology should contain

![Figure 2. Example of the operation of Semantic Web Services](image-url)
a set of concepts, which is complete enough to allow users to present their requests as accurately as possible.

Several XML-based technologies for formulating ontologies already exist, including the OWL Web Ontology Language and the Web Service Modelling Language.

**Semantic Web Services**

Now we know all about ontologies, we are ready to analyse a hypothetical scenario presenting the main concepts behind the operation of the Semantic Web, working from the user's query input to result presentation.

**Phase 1: formulating a goal**

Let us see how our journey planning example could be implemented. Figure 2 presents a summary of all the necessary calls to various Semantic Web Services.

The user is represented by an agent (typically a semantic browser application), which has all the authentication and authorisation data necessary to conduct transactions on the user's behalf. The process starts with the user entering a request, presumably using natural language. A semantic browser then analyses the query using a language parser and stores it using concepts taken from one or more ontologies. The processed query is called a goal. Standardised ontologies are stored in a publicly-available repository. If the user enters a request like *I want to buy an airline ticket from town A to town B, book a hotel in the centre of B and rent a car of make C*, the resulting goal should look something like the structure presented in Figure 3.

The query is only as detailed as the ontology permits, so if the ontology fails to account for a certain feature of an object, then any references to that feature will be ignored. Applying this to our example, if the *car* class in the *journey* ontology does not have a *make* attribute, then we will not be able to specify the make of the rented car in our query. If a website wanted to introduce a car rental service with the option of specifying the make of the car, then the consortium responsible for providing and standardising ontologies would have to extend the *journey* ontology to include this additional semantic element.

**Phase 2: locating services**

Once the goal has been established, the user's agent communicates it to a middle agent. Each processed goal has a corresponding middle agent that has access to the ontology, goal and service description repositories. The goal repository stores the most commonly requested goals, along with additional information, such as the most frequently called services. The SWS repository contains ontology-based semantic descriptions of all registered services, along with statistics detailing their popularity and accuracy. The middle agent has several tasks, including:

- finding services, where the semantic description best matches the user’s goal,
- negotiating transaction details with the user as services are discovered.

Negotiations are necessary in at least two cases. The first such situation is when the goal is too general and several suitable services are discovered (for example, there are many airlines flying from A to B) but no additional parameters were specified (such as the ticket price). In such an event, the user should be presented with a list of matching services so that they can choose the most suitable one.

The other case which requires negotiation is when no registered service fully matches the query requirements. In our example, this might correspond to a situation where the only free rooms are located outside the centre of town B. The user could then be informed of this fact and could decide whether to go along with the suggested transaction modifications or reject them.

Another middle agent task is to create a suitable sequence of service calls (a workflow) to fully serve the user request. This would be necessary whenever no single service exists that provides all the functionality required by the request. Applying this to our example, it is highly unlikely that any single web service would offer a combination of airline reservations, hotel bookings and car rentals in a specified town. Fulfilling the request would typically require calls to three separate services, each charged with handling a specific sub-task.

**Phase 3: calling web services**

Once a middle agent has built a service call workflow based on semantic service descriptions, the time comes to actually call the services. Each service is represented by an agent, and these agents constitute the third layer of communication. Every call to a specific web service starts with the deployment of the service agent within the Semantic Web runtime, and it is only the deployed agent that connects to its assigned service and uses the available interfaces to handle communication with the service. The web service agent can be seen as a very advanced access interface, which *understands* request semantics and uses this information to call the necessary service functions. This architecture means that the Semantic Web does not require web services to be implemented using any specific technology or programming language, but only specifies requirements for ontologies, goals, service descriptions and the three-layered agent model. This leaves web developers free to implement the actual services using any technology they require.

The middle agent constantly controls the execution of the workflow by communicating with agents representing the par-
On the Web

- Adaptive Services Grid Project
  http://www.asg-platform.org/
- OWL Web Ontology Language
  http://www.w3.org/TR/owl-features/
- Web Service Modelling Language
  http://www.wsmo.org/2004/d16/d16.7/v0.1/20040823/
- Semantic network technologies
  http://www.w3.org/2002/ws/Activity/

It is responsible for passing the results from one service to the next within the workflow as necessary and presenting the final results to the user.

Well, that was a rapid introduction to Semantic Web Services. Of course, the multitude and scale of problems still remaining are such that you may well question the plausibility of the whole affair, but let us remain optimistic.

Summary

Interest in the Semantic Web is growing rapidly throughout the world, with the constituent technologies being researched by such giants as IBM and Microsoft. The web service description presented in this article is mainly based on the Adaptive Services Grid specification currently under development by an international consortium as part of the European Union's sixth framework project. The project was officially launched on 31 August 2004 and the expected time required to develop and implement the mechanisms outlined above is five years. Time will tell which ideas will be introduced in the near future, but whatever happens, Europe will not easily give in to US commercial giants.

It is well worth noting that, even though each project takes a slightly different approach to Semantic Web concepts, all proposed solutions have a common denominator involving Java and XML-based languages, so it seems there is no getting away from the growing importance of these technologies in modern IT.

Available XML-based technologies include OWL (Ontology Web Language) for storing ontologies, WSDL (Web Service Definition Language) for describing the functionality and interfaces of web services and SOAP (Simple Object Access Protocol) for exchanging data between various applications and specifying data processing methods. Another universal technology is CORBA (Common Object Request Broker Architecture), which makes it possible to remotely call the methods of server-side objects without having to know the exact location of the server, the implementation language of the methods in question or the operating system they execute in. Each of these technologies is a step towards making the Semantic Web a reality, but we are still missing a central technology, which would take their functionality and mould it into one coherent entity. An interesting candidate is WSMO (Web Service Model Ontology), where the specification covers all the components necessary to build a semantic web, including ontologies, web services, goals and mediators (which make it possible to join ontologies together). The ASG project will be based on the WSMO model.