THE SEMANTIC WEB

Ing. Roman PETRUCHA, Doctoral Degree Programme (1)
Dept. of Information Systems, FIT, BUT
E-mail: petrucha@fit.vutbr.cz

Supervised by: Prof. Tomáš Hruška

ABSTRACT

This paper introduces new trend in Web development called the Semantic Web. We will get familiar with its basic structure and differences between the Semantic Web and current World Wide Web. Rest of paper deals with the technology related to creation of the Semantic Web.

1 INTRODUCTION

The World Wide Web contains huge amounts of information created by many different organizations, communities and individuals for many different reasons. Most of the Web's content today is designed for humans to read, not for computer programs to manipulate it meaningfully. The solution how to develop machine understandable web is to use metadata to describe the data contained on the Web. This new approach is called Semantic Web. The goal of the Semantic Web is to develop enabling standards and technologies designed to help machines understand more information on the Web so that they can support richer discovery, data integration, navigation, and automation of tasks.

2 THE SEMANTIC WEB

The Semantic Web is defined as a representation of data on the World Wide Web. It is not a separate Web, but an extension of the current one, in which information is given a well defined meaning. In this chapter we will get familiar with its basic concepts.

2.1 MAIN PRINCIPLES

Everything can be identified by URI's - Anyone who has control over a part of Web namespace can create a URI (Universal Resource Identifier) and say that it identifies something (people, places, things, etc.) in the physical world.

Resources and links can have types - The resources are Web documents targeted for human consumption and do not commonly contain metadata explaining what they are used for and what are their relationships to other documents. While a knowledgeable human may easily realize what kinds of relationships the resource has (by reading the text around), it is

hard for the machine to make these same decisions. The difference between Semantic Web and current Web is that the resources and links can have types which define concepts that tell a bit more to the machines.

Partial information is tolerated - The Semantic Web as as current Web is unbounded. It sacrificed link integrity for scalability. Authors can easily link to other's resources as they don't have to worry about the links back to their resource. With no way to inform the linkers when the resources are moved we accept that we may get the 404 links (error) informing us that the link no longer leads to some Web resource.

There is no need for absolute truth - Not everything found from the Web is true and the Semantic Web does not change that in any way. Trustworthiness is evaluated by each application that processes the information on the Web. The applications decide what they trust by using the context of the statements (who said what and when and what credentials they had to say it).

Evolution is supported - It is common that similar concepts are often defined by different groups of people in different places or even by the same group at different times. It would often be useful to combine the data available on the Web that uses these concepts.

Minimalist design - Principle of least power: the less rules, the better. The main aim is to standardize no more than is necessary. This means that the Semantic Web is very unconstraining in what it lets say (anyone can say anything about anything). When we started constraining people, they wouldn't be able to build a full range of applications, and the Semantic Web would therefore become useless to some people.

2.2 THE SEMANTIC WEB LAYERS

The basic principles are implemented in the layers of Web technologies and standards. The Semantic Web layer model consists of few layers, which are presented in Figure 1.

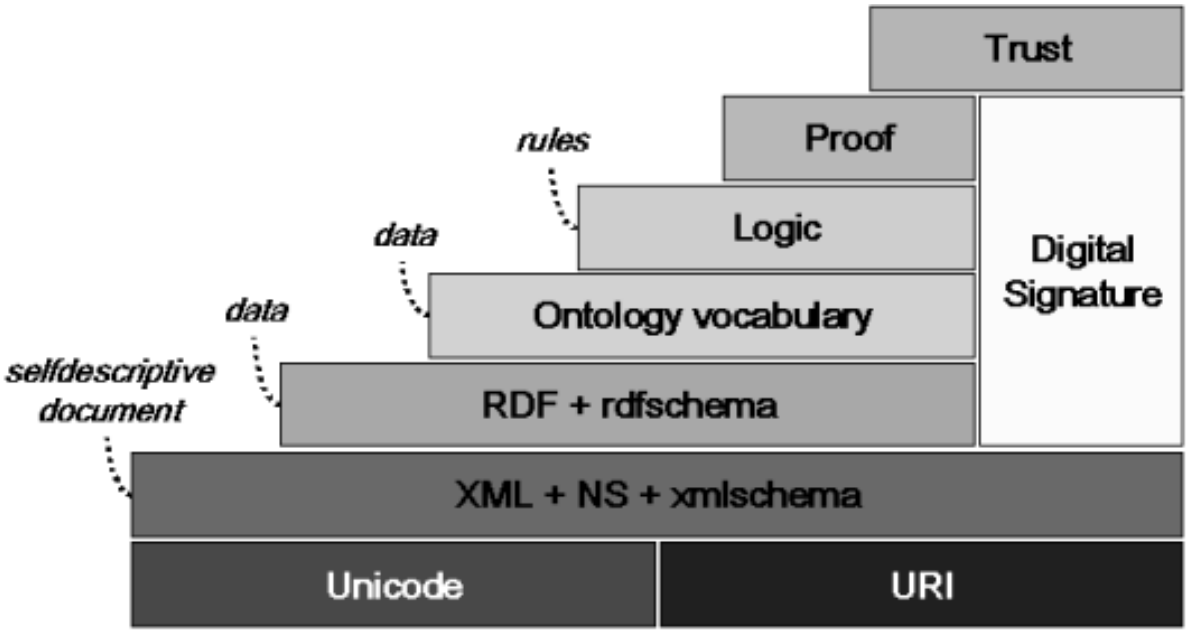

Fig. 1: *The Semantic Web layers*

The Unicode and URI layers make sure that we use international charter sets and

provide means for identifying the objects in the Semantic Web. The XML layer with namespace and schema definitions make sure we can integrate the Semantic Web definitions with the other XML based standards. The RDF layer is the layer where we can give types to resources and links. With RDF and RDFS (RDFSchema) it is possible to make statements about objects with URI's and define vocabularies that can be referred to by URI's. The Ontology layer supports the evolution of vocabularies as it can define relations between the different concepts. The Digital Signature layer is used for detecting alterations to documents. The top three layers Logic, Proof and Trust, are currently being researched. The Logic layer enables the writing of rules. The Proof layer executes the rules and evaluates together with the Trust layer mechanism for applications whether to trust the given proof or not. Application can decide trustworthiness using:

- dependence upon context (trust to verified sources)
- proof checking mechanisms and digital signatures

Digital signatures are simply little bits of code that one can use to verify that he wrote a certain document. We simply apply this technology to RDF. Digital signatures and other proof checking mechanism are currently under development. They will be used in near future. In nowadays it is possible to use only context based mechanism to acquire trustworthiness.

2.3 RDF

The Semantic Web is generally built on syntaxes which use URIs to represent data, usually in triples based structures. Many triples of URI data that can be held in databases, or interchanged on the Web using a set of particular syntaxes developed especially for the task. These syntaxes are called RDF (Resource Description Framework) syntaxes.

When we want to express meaning on the Semantic Web we will need RDF. RDF is a language for encoding knowledge on Web pages to make it understandable to electronic agents searching for information. The main goal of it is to make it possible to specify semantics for data based on XML in a standardized and interoperable manner. Whole RDF is based on XML. It uses XML to exchange descriptions of Web resources, but the resources being described can be of any type, including XML and non-XML resources. Once information is in RDF form, it becomes easy to process it, since RDF is a generic format, which already has many parsers.

The RDF data model consists of three basic object types:

- *Resource* - All things described by RDF expressions are called resources. A resource may be a part of Web page (e.g. HTML element), an entire Web page, whole collection of pages (Web site). For a resource we may consider also an object that is not directly accessible from the Web (e.g. a printed book). Resources are always named by URIs plus an optional anchor id.
- *Properties* - A property is a specific aspect, characteristic, attribute, or relation used to describe a resource. Each property has a specific meaning, defines its permitted values, the types of resources it can describe, and its relationship with other properties.
- *Statements* - A specific resource together with a named property plus the value of that property for that resource is an RDF statement. These three individual parts of a statement are called, the *subject*, the *predicate*, and the *object*. The

object of a statement can be another resource or it can be a literal or a simple string or other primitive datatype defined by XML.

To imagine how the RDF expressions looks like consider following simple sentence:

”*Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila*“

This sentence has these following parts:

- Subject (Resource) ”*http://www.w3.org/Home/Lassila*”
- Predicate (Property) “*Creator*”
- Object (Literal) ”*Ora Lassila*”

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

The RDF used basic serialization syntax. It means that every part of RDF expressions is sorted in specific way into elements and attributes of XML. This is the way how application recognizes what type it is, if subject, predicate or object.

2.4 ONTOLOGIES

Ontology is a term borrowed from philosophy that refers to the science of describing the kinds of entities in the world and how they are related. In Artificial-Intelligence literature is an ontology defined as a specification of a conceptualization (or as formal explicit specification of conceptualization). An ontology defines a common vocabulary for researchers who need to share information in a domain. It includes machine-interpretable definitions of basic concepts in the domain and relations among them.

Basic options of using ontologies:

- support of co-understanding between human
- support of interoperability between computer systems
- simplification of knowledge-based applications design

Practical Applications: Knowledge management in companies, E-Business, Natural language processing, Intelligent integration of information, Resource discovery, Semantic Web Portals, Intelligent education systems

The most important ontology languages are RDFS, DAML+OIL a OWL. There are older languages (e.g. Ontolingua, OCML), but they are used only with older systems. Nowadays it is best to use OWL, new standard recommended by W3C.

2.5 SEMANTIC WEB MINING

If we want the Semantic Web to reach its full potential, many people must start publishing data as RDF. Where is this information going to come from? A lot of it can be derived from many data publications that exist today, using a process called ”screen scraping”. Screen scraping is the act of literally getting the data from a source into a more manageable form (RDF) using whatever means come to hand. Two useful tools for screen scraping are XSLT (an XML transformations language), and RegExps (in Perl, Python, etc.).

However, screen scraping is not often the best solution, so another way to approach it is to build proper RDF systems that take input from the user and then store it straight away in RDF. Data such as you may enter when signing up for a new mail account or buying some CDs online can all be stored as RDF and then used on the Semantic Web.

It will be important to develop tools and easy user interfaces that support users in understanding the metadata and adding the metadata into the Web. This support and automation will be critical in the deployment of the Semantic Web. When more and richer metadata appears there will be huge amounts of opportunities for various applications.

3 CONCLUSION

The real power of the Semantic Web will be realized in future when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs. The effectiveness of such software agents will increase exponentially as more machine-readable Web content and automated services (including other agents) become available. It is only a matter of time until whole current Web will be transformed in the Semantic Web.

REFERENCES

[1] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, May 2001.

[2] Web-Ontology Working Group, http://www.w3.org/2001/sw/WebOnt/

[3] RDF, http://www.w3.org/RDF/

[4] Semantic Web, http://www.w3.org/2001/sw/

[5] The Semantic Web Community Portal, http://www.semanticweb.org/