

DT8107 Distributed Information Systems
**Issues for Utilizing and Enabling
the Global Semantic Web**

Terje Wahl

November 2004

*Department of Computer and Information Science, Norwegian University of Science and Technology
terje.wahl@idi.ntnu.no*

Abstract. Most of the current content of the Web is understandable only to humans. The Semantic Web envisions solving this problem by adding structure and semantic information to the content of Web-pages, enabling machines to understand and reason about information on the web in a generalized manner. This paper presents some important issues that need to be resolved if the vision of the global Semantic Web is to become a reality, mainly focusing on issues related to distributed information systems. The higher layers of the Semantic Web have yet to be fully standardized, and many undecided issues exist within the areas of logic, proof, trust and security. Many of these issues arise due to the heterogeneous, highly distributed and large scale nature of the envisioned Semantic Web. Some important issues that need to be addressed are how to create semantic data sources, robust reasoning, agent coordination, semantic interoperability, trust and security.

1 Introduction

Most of the current content of the Web is understandable only to humans. In addition, it is often difficult to find relevant information when doing searches on the Web. Business applications are abundant on the Web, but when they want to communicate with each other autonomously, they need proprietary interconnections that are relatively difficult to build and maintain.

The Semantic Web envisions solving these problems by transforming the Web into the Semantic Web. This will be done by adding structure and semantic information to the content of Web-pages [1]. This will enable machines to understand and reason about information on the Web in a generalized manner.

There are however many obstacles that need to be overcome before the vision of the Semantic Web can become a reality. This paper will present some of these issues, and look at the present status of some of them. The focus will mainly be on issues relating to the field of distributed information systems.

Section 2 introduces the main Semantic Web technologies currently existing, with subsections on URI, XML, RDF, RDF-S, ontology languages (OWL) and reasoning. Section 3 focuses on issues hindering the utilization of the global Semantic Web, especially within the areas of creating semantic data sources, ontologies, reasoning, agents, trust and security. The conclusion is given in Section 4.

2 Current Semantic Web technologies

The Semantic Web is an extension of the current Web, in which information is given well-defined meaning. This will enable computers and people to cooperate in better ways [1].

The Semantic Web will be highly decentralized because it must be highly scalable to have a global reach. The creation and management of content may be done by almost anyone.

2.1 The Semantic Web tower

The architecture of the Semantic Web can be represented in the form of a tower of specifications and languages layering on top of each other, like in [6]:

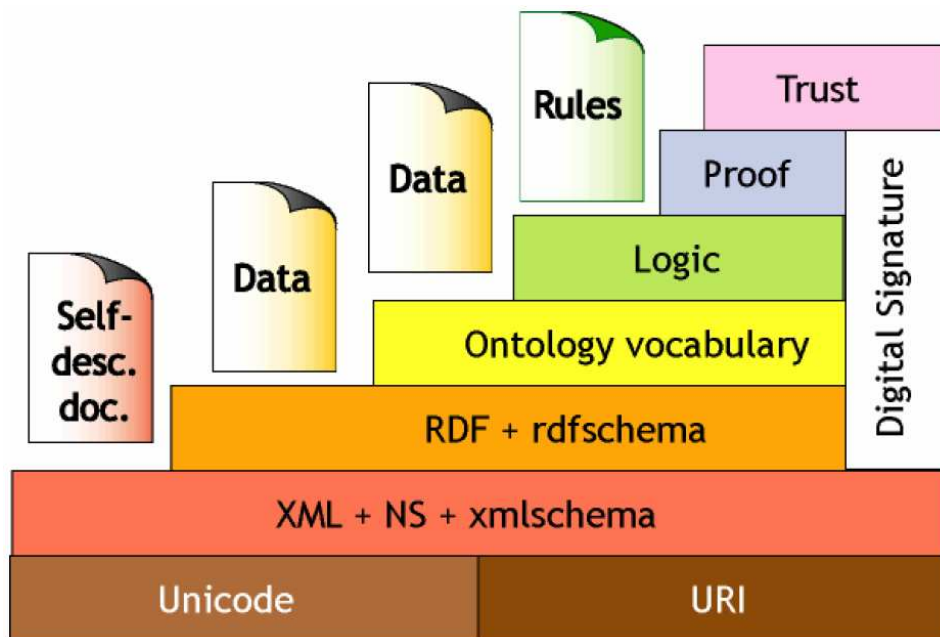


Figure 1. "The Semantic Web Tower" as presented in [6], and originally presented by Tim Berners-Lee at the XML 2000 conference.

On the bottom layer is Unicode and URI. Unicode is a standard for specifying characters and thus create a text. URI is used to specify the location of resources. XML is a standard way of structuring text with self-defined tags. Resource Description Framework (RDF) and RDF-Schema (RDF-S) is based on XML, and allows specifying semantic metadata in a document, e.g. in a HTML-document.

If terms used in RDF-descriptions are explained in ontologies, one can examine the connection between different terms in more advanced ways than with RDF-S. Logic reasoners may be built on top of this to reason on the Semantic Web, so that (more or less robust) proofs can be found when querying the Semantic Web. Trust is an issue on top of all this, because in the distributed and heterogeneous environment of the Semantic Web, it is often important to know who you can trust about the information they provide. Digital signatures should be used across many of the layers in this tower to offer a mechanism of providing trust in a secure way. Security for the Semantic Web in general is an issue that cuts across all the layers [10].

Most of these terms are further described in the subsections below.

2.2 URI

Uniform Resource Identifiers (URIs) are used to locate resources, usually residing on the Internet. All URLs are a subset of URIs. An example of an URL is *http://www.w3c.org*. URIs may also point to a location within a webpage. An example is *http://www.example.org/index.html#section_3*.

2.3 XML

The eXtensible Markup Language (XML) is a meta-language that enables users to create and use their own tags. XML is (like HTML) based on SGML. XML is used to represent these self-defined tags. A Data Type Definition (DTD) is then used to define what vocabulary is legal for the XML-documents that refer to it. XML Schema (XML-S) is used to define the structure of XML documents, and also allows specifying datatypes for values of attributes and the content of elements.

Exchange of XML documents are mostly used for exchanging data with specified syntax. Standard basic XML-documents carry no semantic information [4].

2.4 RDF

While XML is a standard mechanism to structure data, the Resource Description Framework (RDF) is a mechanism to give meaning to data by telling something about it [4].

The RDF data model consists of statements about resources, by using *object-attribute-value* triples. An object is a resource - that is anything that can be identified by a URI, e.g. a webpage, a part of a web-page or a book (URIs doesn't have to specify only resources available on the Web). Values are resources or text strings. For example, the triple "http://www.idi.ntnu.no/~terje, author, 'Terje' " would mean that Terje is the author of the webpage located at the specified URI. In this example, "author" is the *attribute*. Attributes are also called properties. [4]

2.5 RDF-S

RDF-Schema (RDF-S) is an extension that builds on RDF. It allows the definition of classes of resources and domain-specific properties, and can be thought of as a simple type system for RDF [4].

RDF-S allows the definition of class and property hierarchies. It also allows *range* statements restricting legal combinations of properties and classes,

and other statements like the *type* statement, which is used to declare instances of a class. [4]

When combined, RDF and RDF-S can be thought of as a simple ontology language. But it is still limited with regards to areas such as the number of modelling primitives and logical semantics [4]. For this, richer ontology languages should be used on top of RDF and RDF-S, like OWL, which is presented in the next section.

2.6 Ontology languages

In [7], an *ontology* is defined as "a formal, explicit specification of a shared conceptualization". The purpose of ontologies is to explain the understanding of concepts and terms relevant to a given domain [3].

The markup provided by RDF and ontologies enables the creation of systems that can answer complicated questions that cannot be answered by looking at only one webpage [1].

Two examples of ontology languages are DAML and OIL. Features of these two languages have been combined to form the specification for OWL. OWL became a W3C recommendation in February 2004. [11]

2.6.1 OWL

Web Ontology Language (OWL) is an ontology language intended for use on the Web. It is specifically designed for use on the Semantic Web, and uses URI and RDF as a basis.

OWL allows ontologies to be distributed, it scales like the Semantic Web should do, and is an open and extensible standard.

As stated in [11]: "OWL builds on RDF and RDF Schema and adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes."

OWL has three sublanguages: OWL Lite, OWL Description Logics (OWL DL) and OWL Full. These three specifications each add increasing expressiveness.

OWL Lite mainly supports classification hierarchies and simple constraints. OWL DL is as expressive as possible without losing computational completeness and decidability. OWL Full provides great syntactic flexibility

and expressive power, but as a consequence it is probably not possible to create reasoners that provide computational guarantees when using OWL Full.

OWL Full is an extension of RDF-S. OWL Lite and OWL DL are not, because they require restricted use of RDF and RDF-S to be built on top of it.

[11]

2.7 Reasoning

Reasoning on the Semantic Web will be done by inference engines that use logic to process queries and provide (hopefully provable) results to users. Inference engines may be implemented into software agents that act autonomously and communicate with other software agents.

Inference services on the Semantic Web are equivalent of SQL query engines for relational databases, but provide more powerful support [4]. Two examples of inference engines are Ontobroker and FaCT (Fast Classification of Terminologies). These systems help building new ontologies and provide advanced information access and navigation [4].

3 Issues

This chapter goes through some issues that need to be solved before the Semantic Web can be fully utilized on a global scale.

3.1 Creating semantic data sources

Data sources for the Semantic Web may be represented in several ways, but the most common method will probably be enriching HTML-coded documents with semantic information in the form of RDF tags, and some times referring to ontologies contained in external documents.

The effectiveness of software agents that process information from the Semantic Web will increase exponentially as more semantically annotated webpages and automated services become available [1]. At the same time, the incentive to semantically enrich webpages and to create agents depends of the relative immediate usefulness of it. This may create a "vicious circle".

There are several ways to create data sources for the Semantic Web. One might semantically enrich Web-pages by manual adding of semantic

information, but this consumes a lot of time and resources. In addition, there is the issue of user friendliness. Non-technical or untrained users need simple methods and GUI to perform this task. Fully automatic or semi-automatic methods for semantically enriching Web resources are being researched. [5] But this topic is not much concerned with distributed systems, and will therefore not be further discussed in this paper.

3.2 Ontologies

It will be easier to cooperate between agents if a common representation model is decided upon [3]. Currently there exist many standards for writing ontologies. It is possible to implement translations between these languages, but the W3C recommendation of OWL is helpful to speed up the deployment of the Semantic Web.

There is a trade-off between expressive power and efficiency among different modelling languages in large scale distributed systems such as the Semantic Web. It is difficult to support robust reasoning over schemas such as OWL, which has relatively powerful expressiveness. The simpler RDF-standard makes it easier to do reasoning, but is still quite flexible and support rich semantics [3]. Reasoning based on RDF might therefore be an alternative until more robust reasoners for OWL emerge.

3.3 Reasoning

The Semantic Web will be as decentralized as possible. This versatility will lead to paradoxes and unanswerable questions when trying to reason based on the Semantic Web. Semantic Web researchers accept that this is a price that has to be paid to achieve versatility [1]. But this makes it much harder to do robust reasoning on the Semantic Web.

Reasoning on the Semantic Web has not yet been fully realized. Current prototypes seem to compromise on either Knowledge Representation principles and capabilities, or scale and distributiveness [2]. To fully utilize the Semantic Web, one must successfully provide robust reasoning over distributed resources.

Robust reasoning is closely related to *soundness* and *completeness*. First-order logic (FOL) is well established in the field of knowledge representation and reasoning (KRR), and has had some success of providing robust reasoners for these earlier systems. Standardisation of Semantic Web technologies has so far resulted in the ontology-language OWL (among other standards), which is also based on this foundation of FOL [2]. But this might be a problem because of the envisioned size and

diversity of the Semantic Web. Without centralized control, inconsistency will exist. It must probably be accepted that reasoners sometimes will give a wrong answer to a query. Also, it will be hard to require answers to queries to be complete, because we can seldom spend the time or resources to examine *all* the information on the Semantic Web to answer each query. Because of this, incompleteness must probably also be accepted. [9]

Locality is a component of the emergent Semantic Web, because it will be gradually built up from a large number of small local interactions. Many agents operating on the Semantic Web are going to be autonomous and not centrally coordinated. The Semantic Web will also have a lot of randomness, because nodes may change, fail or disconnect at different times. This locality, autonomy and randomness make robust reasoning harder, because they affect global integrity and completeness of the Semantic Web [3].

Automated reasoners must be able to deal with broken or dead semantic links, because they are bound to exist on the Semantic Web with its many diverse information providers. Inference engines must be able to handle this if they are to be successful at providing quality answers to queries.

If we can know which resources we can trust (and hence which resources are the most useful) on the Semantic Web, it would be simpler to build a robust reasoner. For the Semantic Web, this might be implemented by some sort of automated mechanism for trust. The literature suggests that this issue of trust can be solved in combination with the use of software agents [2]. But there are many issues associated with these aspects, as seen in the next chapters.

3.4 Agents

Software Agents will in various forms be used to collect and process information on the Semantic Web [1]. They will be able to communicate and cooperate by exchanging information containing semantic markup [1]. It may however be challenging to accomplish this task. This problem is associated with semantic interoperability, because if two agents with two separate representations of the domain want to communicate, they need some kind of mapping between their vocabularies and their meaning [2]. As stated in [2], "This must be done automatically if the agents are to cooperate autonomously in the Semantic Web".

3.4.1 Semantic Interoperability

There exist several suggestions on how to overcome this problem of semantic interoperability. One solution might be to use a standard upper ontology, or to provide mapping or merging between some terms in two or more ontologies [2].

Several suggestions for such "top-level" ontologies have been created. Examples are Cyc and IEEE's Standard Upper Ontology. These are especially helpful for smaller applications, but if some of these standard upper ontologies were to be used for the Semantic Web, it would limit its distributiveness and decentralisation. This suggests that the use of ontology mappings would be preferable for the Semantic Web. [2]

Ontology mappings may however not be enough to achieve successful knowledge sharing between agents. Erroneous results may be produced if the inference engines of two communicating agents are based on different logic systems, e.g. inference logic versus relevance logic. Such an error may therefore arise even if the agents use the same ontologies on their knowledge bases. [2]

A suggested solution to this issue is that agents negotiate or at least communicate which logical system is used. Another suggestion is to use an information-theoretic approach, e.g. that of Barwise and Seligman's channel theory. Their suggested approach takes into account different understandings of semantic terms, and allows ontology mappings explaining different use of terms in different contexts related to local concepts. [2]

3.4.2 Agent communication infrastructure

To be able to communicate and reason about semantic information, one must be able to locate resources at the network. The mechanism for doing this should be decentralized, since the Semantic Web itself aims at being so, to facilitate good scalability. [3]

There are three main directions of design in P2P-systems to do decentralized resource location [3]:

1. Unstructured P2P-systems that are based on gossiping techniques.
2. Hierarchical P2P-systems that have specialized superpeers responsible for routing.
3. Structured P2P-systems based on distributed hash tables.

The third approach combines efficient search and maintenance without using centralized components. But it is still an open issue what method(s) will be used for the Semantic Web in the future. Load balancing is also an undecided issue regarding these P2P-systems. [3]

3.5 Trust

Traditionally, trust has not been an issue in knowledge-based systems, because these systems have not been highly distributed. But on the Semantic Web, where almost anyone or anything can be a source of information, trust becomes an issue [2]. [8] asks: "How can trust be modelled and exchanged between agents and Semantic Web services? Where should trust annotations be stored and made available? What kind of knowledge is required to measure trust and where will this knowledge come from? What trust features need to be considered (e.g. subjectivity, propagations, transitivity)? And how do they affect trust in general?"

Some work has been done to create a conceptual model of trust [2]. This is helpful, but how should this be used to implement trust in the Semantic Web? This is still an open issue, but there exist some suggestions. One approach is to implement trust as an addition to an existing system, but this might not be very efficient. It is argued that trust needs to be added to the system design process, and that "there is a clear need for semantic language support for representing trust" [2]. There exist some emerging standards (such as SOAP security extension), but none of this work has reached far enough to be able to represent trust in an efficient manner [2].

3.5.1 The source of trust information

It is not yet decided if trust should be centrally controlled and managed, or if it should be distributed such that many agents may contain trust information and share this among them. There have been several prototypes that suggest some kind of centralized server for storing trust information. This raises the questions if the agents or users providing information regarding trust to this service can be trusted, and if the trust service itself can be trusted. Another issue is that centralized systems in a highly distributed setting tend to have limitations to scalability. [2]

3.5.2 How to measure trust

The question of how to measure trust is still an open issue. Such a measurement could be based on the experience of agents, but the concrete measure of this and how this information can be distributed is still an open issue [2].

Also, when you want to measure trust in a diverse system like the Semantic Web, trust needs to be measured in a context. For example, you may trust an agent to recommend a good restaurant, but not to perform payment services on your behalf. This makes the measure of trust even more complex and harder to compute. [2]

3.6 Security

Little research has been conducted on security for the Semantic Web, in contrast to many other areas within the Semantic Web [10]. There is a need to establish methods to avoid unauthorized access and malicious modifications on the Semantic Web. Privacy also needs to be retained. In [10] it is argued that security needs to be inserted into the system from the beginning, and not added to it afterwards.

If the Semantic Web is to be secure, all its layers (as shown in Figure 1) need to be secure. Secure interoperability must also be ensured. There currently exist security mechanisms for TCP/IP, sockets and HTTP. But for the higher layers of the Semantic web, security is still an open issue. XML needs to be secure, for example by controlling access to various portions of a document. RDF needs to be secure with regards to interpretations and semantics, because one might want to specify security relative to a context. Ontologies should be able to have security levels attached to them. Secure reasoning by information integration is a big challenge. As stated in [10], information on the Semantic Web should be "managed, integrated and exchanged securely". [10]

Privacy is a closely related issue. Some documents or parts of them may be private, while other parts may be public or partially public. It is an important issue to examine how the Semantic Web may reach its visions without losing the ability to maintain privacy and anonymity when wanted [10]. If no restrictions are applied to security or privacy on the Semantic Web, it is easy to imagine possible abuse of private information if it is semantically enriched and searchable by anyone on the distributed Semantic Web.

4 Conclusion

The basic layers of the Semantic Web contain well specified standards, like Unicode, URI, XML, RDF and RDF-S. OWL has appeared in 2004 as a W3C recommendation for an ontology language designed for use with the Semantic Web.

The higher layers of the Semantic Web have yet to be fully standardized, and many undecided issues exist within the areas of logic, proof, trust and security. Many of these issues arise due to the heterogeneous, highly distributed and large scale nature of the envisioned Semantic Web. Some important issues that need to be addressed are how to create semantic data sources, robust reasoning, agent coordination, semantic interoperability, trust and security.

It may take some time to resolve all these issues, but if the Semantic Web is realized according to its vision, it will be a revolutionary and powerful tool to search for and share semantically enriched information on a global scale, processable by both humans and machines.

5 References

1. Berners-Lee, Hendler, Lassila: *The Semantic Web*. Scientific American 2001.
2. Kalfoglou, Alani, Schorlemmer, Walton: *On the Emergent Semantic Web and Overlooked Issues*. ISWC 2004.
3. Karl Aberer et al.: *Emergent Semantics Principles and Issues*. DASFAA 2004, 2004.
4. Ding, Fensel, Klein, Omelayenko: *The Semantic Web: Yet Another Hip?* Data and Knowledge Engineering, 2002.
5. Rinaldi, Kaljurand, Dowdall, Hess: *Breaking the Deadlock*. CoopIS/DOA/ODBASE 2003, 2003.
6. Patel-Schneider, Fensel: *Layering the Semantic Web: Problems and Directions*. ISWC 2002, 2002.
7. Gruber: *A Translation Approach to Portable Ontology Specifications*. Knowledge Acquisition, 1993.
8. Matthews, Dimitrakos: *Deploying trust policies on the Semantic Web*. iTrust'04 2004.
9. Harmelen: *How the Semantic Web will change KR: Challenges and opportunities for a new research agenda*. The Knowledge Engineering Review, 17, 2002.
10. Ferrari, Thuraisingham: *Security and Privacy for Web Databases and Services*. EDBT 2004, 2004.

6 Additional resources

11. <http://www.w3c.org>.